611-TD-569-001

EOSDIS Core System Project

M&O Procedures: Section 11 — Production Rules for PGE's to Function in ECS

Interim Update

March 2000

Raytheon Systems Company
Upper Marlboro, Maryland

Preface

This document is an interim update to the Mission Operations Procedures Manual for the ECS Project, document number 611-CD-004-004. This document has not been submitted to NASA for approval, and should be considered unofficial.

This document has been revised to meet The Drop 5B Science System Release Plan.

Any questions should be addressed to Elroy McLeod or Rodney Creecy:

Data Management Office
The ECS Project Office
Raytheon Systems Company
1616 McCormick Drive
Upper Marlboro, Maryland 20774-5301

This page intentionally left blank.

11. Production Rules for PGE's to Function in ECS

11.1 Production Rules

11.1.1 Purpose and Scope

This section describes the production rules supported by the Release B CDR design. It is intended to explain the implementation of these production rules in ECS and to document the information required for each rule. This section does not cover the exact design and implementation of these production rules in the PDPS system. That information may be found in the PDPS design documents, primarily in the Planning Subsystem document 305-CD-026-002, Release B SDPS Planning Subsystem Design Specification for the ECS Project.
. While this section does address the syntax and operational procedures for specifying production rules, it does provide detailed information about what data is required for each rule and how that data will be used

11.1.2 Overview of Production Rules

11.1.2.1 Data Processing in ECS

Before discussing production rules, it is important to have a basic understanding of how the Earth Observing System Data and Information System (EOSDIS) Core System (ECS) conducts its data processing. ECS provides facilities for running product generation executives (PGEs) to produce science data products. The support for this function primarily resides in three subsystems: the Data Processing Subsystem (DPS) which actually executes the PGEs, the Planning Subsystem (PLS) which plans the execution of PGEs to efficiently utilize the local computing resources and the Data Server Subsystem (DSS) which provides PGEs with input data and archives output data. The design and implementation of the DPS and the PLS are carried out by the ECS organization known as the Planning and Data Processing System (PDPS).
Data Processing Subsystem (DPS) - The DPS provides the environment under which PGEs are run. It works with the SDP toolkit to interface with the science software. The DPS coordinates the acquisition of input data (staging) and the archiving of output data (de-staging) with the DSS. The DPS also ensures that there are adequate resources (disk space, RAM, etc.) available before a PGE begins execution. The DPS uses the AutoSys COTS product to implement the PLS production plan.
Planning Subsystem (PLS) - The PLS generally manages the ECS data processing activities and provides the main interface between the DPS and the rest of ECS. The PLS plans the execution of PGEs to provide efficient use of computing resources. The PLS receives. Production Requests (PRs) either from a software tool (Production Request Editor) or, for on-demand production requests (OPRs), from the DSS or DPS (see Error Handling). The PLS breaks up PRs into data processing requests (DPRs), which are individual runs of a PGE. It maintains information about input data for specific DPRs and passes that information to the DPS. It queries the DSS for data which meets required criteria and provides the universal references (URs) of that data to the DPS for acquisition.
Data Server Subsystem (DSS) - The DSS provides input data to the DPS and archives output data from the DPS. It responds to queries from the PLS and provides the PLS notification of the arrival of input data for all subscriptions the PLS has placed with it. The DSS also submits On-Demand

Production Requests (OPRs) to the PLS.
These three subsystems form the core of the ECS data processing services. Since it has responsibility for managing these activities, most of the information about production rules will be maintained by the PLS.

11.1.2.2 General Definition of Production Rules

Simply stated, production rules are the instructions about how a particular PGE is to be run.These instructions specify a wide range of information such as the input and output data types, the frequency of execution, activation conditions and error handling instructions. This section focuses on the other types of production rules such as alternate inputs to be used or conditional PGE activations. These and other production rules are defined in the following sections.
Production rules in the ECS system will be tied to PGEs. This means that each PGE will use one or more production rules. The production rules will be initially entered by the Instrument Team with further modifications if necessary when a PGE undergoes Science Software Integration and Test (SSI&T) at the DAAC. Where applicable, default parameter values will be entered at that time. Many of these defaults can be overridden in the production environment when a Production Request is entered.

11.2 Production Rule Definitions

Introduction

This section presents a basic definition and examples of each of the production rules currently supported by the Release B CDR design. The production rules are divided into three categories:

Input Data Specification
Conditional Activation
Error Handling

.Input Data Specification rules specify how the inputs for a particular run of a PGE are identified. Conditional Activation rules stipulate the conditions of when a particular PGE is to be run. Error Handling rules specify automatic actions to be taken when a PGE fails. Each sub-section is titled with the name of the production rule being discussed.
It is hoped that providing a common nomenclature will facilitate any future discussions between ITs and ECS about production rules. After each rule there is a list of the information required to implement that rule in the production environment (i.e. post-launch, at the DAACs). This information will initially be provided at SSI&T time. In the operational environment many of the values can be set when a Production Request is entered. Most production rule sections also include a simple diagram to help illustrate the concept being presented.

11.2.1 Input Data Specification

These production rules generally stipulate how inputs for a particular run of a PGE (a DPR) are identified/selected/created. Most PGEs have at least one input and one output. The most basic information about these inputs and outputs is the Earth Science Data Type (ESDT). It is assumed in the examples in this chapter that the ESDT is specified for all data sets which are used or produced by a PGE.

11.2.1.1 Basic Temporal

The most basic type of production rule specifies, for each input ESDT, the temporal range of that ESDT required by the PGE.
For example, the run of a PGE which produces a particular 2.5 minute MODIS Level 1B data granule requires as input the specific 2.5 minute Level 1A granule which is ontemporaneous with the Level 1B granule to be produced.
Another example would be the production of a CERES Level 1A granule from the Level 0 data. CERES L1A granules cover 24 hours and require the PLS to identify and the DPS to stage all of the Level 0 data files that cover a particular 24 hour period. This sort of simple temporal specification is basic to the system and will be used by every PGE.
Inputs required to implement the Basic Temporal rule
For each PGE which uses this rule, the following data must be provided for each ESDT (input oroutput): Time period - Length of time of ESDT (e.g. 12 hours, 1 week or one year).
Boundary - Date/time to start counting periods
Both of the above parameters are specified at SSI&T time.

11.2.1.2 Example use of the Basic Temporal Rule

Suppose there is a PGE which creates a daily (24 hour) data set and those data sets should correspond with calendar days. In that case Time Period = 24 hours and Boundary = 1/1/98 00:00 (this could be any date). When a Production Request is entered, the Planning Subsystem will scan backwards from the start time of the PR until it finds the start of a period. It then produces DPRs for each period of time covered by the PR. Table 11.2.1.2-1 illustrates how this

would work. Given the period and boundary defined above, two PRs are entered, the first for the time between 6/3/99 12:00 - 6/7/99 6:00 while the second specifies 6/7/99 18:00 - 6/16/99 12:00. When the first PR (#1) is entered, the Planning Subsystem determines that the start time of the PR is not the start time of a period. It then scans backwards, finding the start time of the period to be 6/3/99 00:00 and creates a DPR (1.1) to process that period. It then continues producing DPRs (1.2, 1.3, 1.4 and 1.5) until it reaches the end of the period which contains the end time of the PR. When the second PR (#2) is entered, the same process takes place. Note that since the end time of PR #1 and the start time of PR #2 fall in the same period, duplicate DPRs (1.5 and 2.1) are created. This sort of duplication is checked for by the Planning Subsystem and the second DPR is not run.

Boundary (1/1/98 00:00)
= Period (24 Hrs)
PR #1 PR #2
DPR 1.1
DPR 1.2
DPR 1.3
DPR 1.4
DPR 1.5 = DPR 2.1
DPR 2.2
DPR 2.3
DPR 2.10
DPR 2.9
6/1/99 6/23/99

Table 11.2.1.2-1. Basic Temporal

11.2.1.3 Alternate Inputs

This is a fairly simple rule, although there are several options which give it great flexibility and power. There are cases where, for one of its input data sets, a PGE can use any one of several inputs. For example, a particular PGE might require model wind data as an input and would be capable of accepting wind data from a DAO model, an NCEP model or, as a last resort, could use climatology. Variants of this rule allow the alternate input to be optional or allow alternates to be grouped so that more than one of the alternates may be needed (i.e. use any M of N ESDTs in the group).

11.2.1.4 Inputs required to implement the Alternate Inputs rule

ESDT of Primary Alternate - This is the data set ID for the most desirable alternative.
Timer for Primary Alternate - Amount of time to wait before attempting to use second alternate.
Number Needed - Number of the alternate data sets (from this group of alternatives) required by the PGE. Usually 1, but can be more. If set to 0, this set of alternates is optional (i.e. the PGE can run without any of the alternates) ESDT of Second Alternate -
Data set ID for Second Alternate
Timer for second Alternate - Amount of time to wait before attempting to use third alternate..
(include any number of alternates, no practical limit).
ESDT of Last Alternate - Data set ID for Last Alternate.
Timer for Last Alternate - Note that this timer is used only if Number Needed = 0, which indicates that this set of alternates is optional. In that case, if this timer expires, the DPR will be

executed even if none of the alternates is available. If however, Number Needed > 0, then this timer is not used; the DPR will be held indefinitely, waiting for one of the alternates to become available.

11.2.1.5 Example use of the Alternate Inputs Rule:

Let's assume that a PGE has two required input data sets and a third data set which can be any one of three alternates. Of the three alternates, the first two choices are almost equivalent, while the last choice is less desirable. In this case we designate the Primary Alternate (first choice) and set a timer for 1 hour. The second choice is designated with a timer set for 4 hours. This situation is shown in Table 11.2.4-1. What will happen is that, after the two required inputs are available, an attempt is made to acquire the first choice. If that choice is unavailable, the Planning Subsystem holds the DPR and starts the first timer (1 hour). If the first choice becomes available in that hour, the DPR is started immediately. If the first timer expires and the first alternate is still unavailable, an attempt is made to use the second choice. If the second choice is unavailable, the second timer (4 hours) is started. If the first or second alternates become available at any time during that 4 hours, the DPR is started immediately. However, if the second timer expires (a total of 5 hours after the two required inputs are available), an attempt is made to use the third choice. Unless this set of alternates is optional, most PGEs are expected to have a last alternate which is always available (e.g. climatology). If not, the DPR is held indefinitely waiting for one of the alternates to become available.

Required Dataset 1
Primary Alternate
Second Alternate
DPR
< or > First Choice –Timer set to 1 hour.
Wait 1 hour after required data sets are availabe before trying to use second alternate.
Second Choice –Timer set to 4 hours.
Wait 4 hours after this choice was first tried before attempting to use third alternate.
Output Dataset
Third Alternate < or > Third (in this case, last) choice. Attempt to use if primary and secondary alternates are unavailable 5 hours after required datasets are available.
Required Dataset 2

Table 11.2.1.5-1. Alternate Inputs

11.2.1.6 Tiling

In this case a PGE is set up to run for a series of pre-defined, rectangular areas (tiles). The tile definitions need to be entered into the PDPS database so that the proper input data granules can be identified. The implementation of this rule calls for the PGE developers to create tile definition files which would have descriptions of each tile. The Planning subsystem would then use those definitions to query the Data Server for input data granules for each tile. Tiles are grouped together into clusters
which are likely to share common input granules. This is done for efficiency of operations.

11.2.1.7 Inputs required to implement the Tiling rule:

The primary input for the tiling rule is a tile definition file. This file will contain one entry for every tile which will include: Tile ID - Unique identifier used to refer to the tile Lat./Lon. of tile

corners. Four coupled pairs of coordinates which bound the tile. Cluster ID - ID of cluster in which tile falls.

Tile 1 Input Data
Tile 2 Input Data
Tile 3 Input Data
Tiles 4-6
Tiles 7 and 8
DPR-1
DPR-2
DPR-3
Output granule for tile 1
Output granule for tile 2
Output granulefor tile 3
Tiles 4-8 are processed in the same manner as tiles 1-3.
8 Tiles clustered together since they are likely to have overlapping inputs
Original
Swath
Data

Table 11.2.1.7-1. Tiling

11.2.1.8 Example use of the Tiling Rule

A tile definition file is part of the package delivered with the PGE for SSI&T. In the production environment, a PR is entered to run the PGE on all the tiles. A DPR is generated for each tile. These DPRs are grouped by Cluster ID so that, since they will share some of their inputs, these DPRs will be run on the same machine which will minimze the the staging and copying of those inputs amongst the various production machines. The PLS will use the tile definitions to query the DSS to idenify all input data granules covering a tile. It is likely that many of these input granules will have data outside as well as inside each tile. The URs for these input granules and the specific tile definition are then passed to the DPS as part of each DPR. Each DPR will then run, the DPS will aquire the inputs from the DSS and the PGE will run and, using only the data in the tile, create an output data granule for its tile.

11.2.1.9 Data Server Proxy (Subsetting/Subsampling)

There are cases where a PGE would like to use Data Server services on its input data sets prior to running the PGE. In these cases the Planning and Data Processing subsystems simply act as proxies for the PGE by calling those Data Server services. The services provided here are totally dependent what services the DSS offers for each specific product. Instrument teams wishing to use specific services on specific products should coordinate with the Data Server Subsystem to ensure those services will be available.

11.2.1.10 Inputs required to implement the Data Server Proxy rule

Each PGE which uses this rule will be need to identify:
Input ESDT - ID of data set upon which Data Server services will be performed.
Service Requested - Data server service ID Interface Parameters - Parameters specific to the service being requested. For example, a spatial subset request would specify the geographic extent while a temporal subset request would specify the time period requested.

Subset of Original Dataset
DPR Output Data Granule
DPR Output Data Granule Subsample of Original Dataset

Spatial Subsetting Subsampling

Table 11.2.1.9-1. Subsetting & Subsampling

11.2.1.11 Example use of the Data Server Proxy Rule

Assume a PGE with an input data set which covers the entire globe and is gridded at a .1 o by .1 o resolution. During the post-launch, PGE shakedown phase, the science software developer might want to generate a 1 o by 1 o resolution product for a quick look at the output data. Rather than have the PGE do its own averaging/subsampling, Data Server subsampling services could be invoked so that the PGE only has to run on 1/100th the amount of the original data. In this case a second PGE profile would be created at SSI&T which would include the service request and track the different runtime characteristics (e.g. CPU time, disk storage for input/outputs, RAM usage, etc.)

11.2.1.12 Level 0 to Level 1A

This is a special case of the simple temporal range rule and is of interest only to certain Level 1A PGE developers. The issue is what rules PDPS uses to determine what Level 0 data files are required to produce a particular Level 1A granule. In many cases (such as CERES), this is handled by the Basic Temporal rule rather than the method described here. This rule is intended to handle orbit-based Level 1A granules. Level 0 'granules' received by EDOS will be not have any discernible relationship with the spacecraft orbit. The Level 0 'granules' have not even been fully defined for some instruments flying on other platforms. In these cases, the Planning Subsystem will access a crude orbit model (really just a lookup table) to determine the start/stop times of a given orbit. It will then use those start/stop times (after adding a small cushion to ensure complete coverage of an orbit) to query the data server for the Level 0 data covering the orbit. The response to that query is given to the Data Processing Subsystem which acquires the data. The PGE then uses SDP Toolkit calls to determine the exact extent of the orbit. A similar process will be used for instruments having orbit-based Level 1A granules but which will be flying on platforms other than AM-1. In the case of SeaWinds, flying on the ADEOS II platform, a temporal offset may have to be used if the orbit model is based on equator crossing rather than on pole crossing as the SeaWinds want for their Level 1A granules.
Time

Time
Level 0 Data
Orbit-based
Level 1A Granules
Using temporal definitions of orbits, the PLS identifies and the DPS stages the appropriate Level 0 Data for each orbit-based Level 1A granules

Table 11.2.1.12-1. Level 0 to Level 1A

11.2.2 Conditional Activation:

Most PGEs have well defined times and conditions when they are to be executed. The most common activation condition is the availability of all input data sets. Similarly, the frequency of execution is usually well defined (e.g. run once for every granule or run monthly averages once a month). However, some PGEs might have additional/different constraints on when they get run. This section addresses those cases.

11.2.2.1 Intermittent Execution

A PGE can be set up to run on every Nth instance of input data. For example, a QA PGE that is run on a daily product may only need to be run every fifth day to provide a spot check. Note that this does not refer to the common case of only running a weekly averaging PGEs once each week, which would be handled by the Basic Temporal rule and the time ranges specified for the input and output ESDTs. Rather this is a special case where a PGE can be run every day (or hour, week, etc.), but, for whatever reason, it is only desired to be run every Nth day.

11.2.2.2 Inputs required to implement the Intermittent Execution rule

This rule is implemented by using two parameters:
Number to Skip - Number of DPRs to be skipped (not executed)
Number to Keep - After skipping the specified number of DPRs, how many are to be kept. This number will usually be one, but could be any number. The use of these parameters will allow a pattern of execution to be established. This pattern is not maintained between PRs.

PR #2
PR #1
DPR
1.1
DPR
1.2
DPR
1.3
DPR
1.4
DPR
2.1
DPR
2.2
DPR
2.3
DPR
2.4
DPR
2.5
DPR
1.2
DPR
1.3
DPR
2.3
DPR
2.2
DPR
2.5
PRs #1 and #2 are entered at different times.
PR #1generates 4 DPRs and PR #2 Generates 5.
Only 5 of the 9 are ‘kept’and actually run.
Number Skipped = 1

Number Kept =2
Created Run
Seperate PRs,
pattern is restarted for PR#2

Tabe 11.2.2.2-1. Intermittent Execution

11.2.2.3 Example use of the Intermittent Execution Rule

Using the above example of a QA PGE to be run every fifth day, let's assume a Production Request is entered which covers the period from 6/1 - 6/30. As part of the PR values are entered for number skipped (4) and number kept (1). The PR would be expanded into 30 daily DPRs, of which, four out of every five would be discarded, leaving one DPR every fifth day.

11.2.2.4 Metadata-based Conditional Activation

It is possible to determine if a given DPR should be run, based on the metadata of one or more of its input data sets. For example, a PGE could be set up so that a QA flag must be set to an acceptable level/state within the metadata of an input data set or the PGE should not be run. This production rule will work for both core and product-specific metadata. Note that this is different than data-based conditional activation, which will not be supported in Release B. If that sort of conditional activation is desired, the IT will need to define a product-specific metadata field which will be filled by the PGE producing that data.

11.2.2.5 Inputs required to implement the Metadata-based Conditional Activation rule

This rule is implemented by using a series of statements for each ESDT to be checked. These statements take the basic form of:
Metadata field Operand Value. These statements would be 'AND'ed together so that if any of the checks fail, the DPR will not be run. These statements would be entered in at SSI&T time, however, the values could be changed when a Production Request is entered.

Input 1
QAFlag=7
DayNight=‘DAY’
Clouds=20%
Input 2
QAFlag=4
Input 1
QAFlag=9
DayNight=‘DAY’
Clouds=70%
Input 2
QAFlag=8
Input 1
QAFlag=7
DayNight=‘DAY’
Clouds=30%
Input 2
QAFlag=7
DPR 1

DPR 2
DPR 3
Metadata Checks:
Input 1:
QAFlag > 5
DayNight = ‘Day’
Cloud < 40%
Input 2:
QAFlag > 6
DPR 1 is NOT run since Input 2 failed the QAFlag check.
DPR 2 is NOT run since Input 1failed the cloud cover check.
DPR 3 IS run since Inputs 1 and 2 met all metadata conditions

.Table 11.2.2.5-1. Metadata-based Conditional Activation

11.2.2.6 Example use of the Metadata-based Conditional Activation Rule

Assume there is a PGE which uses multiple ESDTs as inputs. Two of the inputs (Input 1 and Input 2) need to have their metadata checked prior to the PGE being executed. Input 1 needs to have a certain quality, less than a certain percentage of clouds and be daytime data while Input 2 just needs to be of a certain quality. Table 11.12.3.4-1, illustrates this situation and shows three DPRs created for different instances of Inputs 1 and 2 and the disposition of those DPRs based on the metadata of the inputs. As stated earlier, if it was decided that the cloud cover rule was too strict, the value used for comparison could be changed when Production Requests are entered. If, however, a new check were needed for some other metadata field, this change would have to be done through SSI&T.

11.2.2.7 Mode-based Conditional Activation

In this case, the mode a given instrument is in will determine which PGE is run. For example an instrument might go into a calibration mode which requires that a special calibration PGE is run. Actually, this is just a specialized case of the metadata-based conditional activation. The added functionality here is that at SSI&T time PGEs can be grouped into PGE collections. In such cases, the instrument mode would determine which PGE in the collection is run. This mechanism is used primarily to improve the accuracy of plans generated by the PLS .

11.2.3 Error Handling

Error handling is a little different from the other two categories of production rules. This is because it is mostly an operational procedure which occurs in response to an (hopefully) anomalous event. The key mechanism for implementing error handling will be PGE exit codes. Release A PDPS is currently working to define PGE exit codes and determine how best to associate actions with exit codes. (Establishing Science Software Exit Conditions for the Production Environment, 420-WP-006-001) is in preparation. The key enhancement to error handling by Release B is the reuse of the on-demand processing mechanism to automate the response to a specific error code. In this case, when a PGE undergoes SSI&T, On-Demand Production Requests (OPRs) are then associated with various PGE exit codes.

11.2.3.1 Inputs required to implement the Automated Error Handling

At SSI&T time, for every exit condition which will need another PGE to be run, the following information will need to be entered:

Exit Code - PGE exit code which triggers action Message - Message to be displayed to operation console OPR - On-Demand Production Request to be executed. The OPR will have the same timeframe as the original DPR so if the DPR were run on data from 6/20/99 12:00-13:00, then the OPR would be given the same timeframe before being passed to PLS.

11.2.3.2 Example use of Automated Error Handling

It is difficult for a short example to capture the full error handling options of the ECS production system. The following example simply gives a flavor for what is possible with regard to automatically submitting OPRs based on a PGEs exit code. While a certain PGE (PGE1) is undergoing SSI&T it is set up so that if PGE1 has an exit code of 6, it should be re-run in debug mode. If it has an exit code of 12 (can only happen from debug mode), then a second PGE (PGE2) should be run. At a later date, in the operational environment, a DPR for PGE1 is running and terminates with exit code 6. An OPR is generated for PGE1 which includes the runtime flag used to send it into debug mode. A message is displayed on the operator's console informing them of this event. The OPR is run and it finishes with an exit code of 12. Now another OPR is created, this one for PGE2. A different message is displayed on the operator's console.

11.3 Combinations of Production Rules

Introduction

One of the most powerful features of production rules is the ability to combine multiple rules. This gives the science software developers greater flexibility and control over the production of their data. This section is intended to illustrate how some of the production rules can be used together. It is not intended to be exhaustive, but rather to give a sampling of the more common combinations. Most combinations are quite easy to understand and the implications of these combinations are quite clear. However, while theoretically any and all of the production rules can be combined, some combinations will make little sense. For instance, combining tiling with intermittent execution will only produce DPRs for some tiles and it will not be easy for the science software developer to determine those tiles in advance. Conversely, intermittent execution works quite well with the basic temporal rule and behaves in an easily predictable manner. While the software can handle complex combinations, in practice these combinations might have results which are not especially intuitive. For example, while combining alternate inputs with metadata based activation, tiling and intermittent execution would seem quite reasonable to the Planning Subsystem software, it would be difficult for a human to determine the results in advance. However, if intermittent execution is removed from the above combination, the remaining combination might be a perfectly valid production recipe. The following sections provide a few examples of how production rules may be combined.

11.3.1 Basic Temporal

The Basic Temporal rule is fundamental to the production system. All Production Requests will have a temporal component. Consequently, all of the other production rules are being combined with the Basic Temporal rule.

11.3.2 Alternate Inputs and Metadata-Based Conditional Activation

It is possible to combine these two rules so that the metadata of the inputs determines which alternate is used. For example, suppose there is a PGE which, along with its required inputs, can use one of two alternates, but the primary alternate must have a certain level of QA set in its metadata. If the primary data set becomes available, its metadata is checked and, if it fails the check, an attempt is made to use the second alternate.

11.3.3 Alternate Inputs and Data Server Proxy

This combination if quite straightforward. After the input data sets are determined via the Alternate Inputs rule, the Data Server Proxy service is invoked.

11.3.4 Alternate Inputs and Tiling

In this combination determining which data granules fall within a tile is a completely separate activity from determining which alternate is most should be used. In this case the tile definition could be used to acquire the data sets which had be chosen as part of the alternate input process.

11.3.5 Intermittent Execution

While it was mentioned above that all PGEs use the Basic Temporal rule, it is worth emphasizing that point in the case of Intermittent Execution. This rule is somewhat unique since it doesn't create DPRs, it removes them. By definition this rule needs to be used in combination with another rule.

11.3.6 Changes to rules

The following changes have occurred in the listing and organization of the rules documented by this section:
· The Release A Basic Temporal rule has been added.
· The current Alternate Inputs rule is the consolidation of Optional Inputs, Alternate Inputs - Hierarchical Preference and Alternate Inputs - No Preference. Subsetting and Subsampling have been combined and renamed Data Server Proxy· Alternate Inputs - Temporal/Spatial Tradeoff has been left out. This is because the PDPS team is still determining the best manner to implement this rule. There are several reasons for this including the lack of an IT provided algorithm, the nature of which could impact the design and implementation. Depending on what variables the algorithm uses, this rule might actually be a variant of the Data Server Proxy rule, or it might require the algorithm(s) to be integrated into PLS code.

11.4 Production Rules Technical Information Sources

1 ECS Baseline Information System :

2 PDPS Home Page: http://dmserver.gsfc.nasa.gov/ecsdev/relb/pdps/index.html

http://pete.hitc.com/baseline/ -choose drop4, you will find latest

patch documentation and much more.

3 EOS Instrument Team Science Software (PGE's):

http://ecsinfo.hitc.com/iteams/Science/science.html. Production Rules White Paper and much more.

4 MODIS - Science Data Processing software Release 4 System Description, SDST-104 dated August 25, 1998.

5 Test Scenarios for selected Production Rules can be viewed by accessing the SCF at: **</home/dheroux/DPS/TESTBED/MISR/SSIT/V2/ODL/Scenarios**>

11.4.1 Production_Rules_Syntax

The Production Rules Syntax are presented as part of the Powerpoint Slides that accompany this document.

Production Rules identified thus far are listed as follows:

Basic Temporal, Advanced Temporal, Period Specification, Alternate Inputs, Optional Inputs, Metadata-based Activation, Metadata-Based Query -Static, Metadata-based Query - Dynamic, Orbit-Based Activation, Orbit Path, Runtime Parameter, Multi-File Granules, Multi-Granule ESDT's, Spatial Query, Minimum Number of Granules, Land Tiling, Tiling with Metadata-based Query, Optional DPRs, Ocean Data Day, Most Recent Granule, Alternates based on Minimum number of Granules, Zonal Tiling, Tile Clustering and Smart_Start_of_Year.

11.5 Production Requests

11.5.1 Science Software and Production Requests

Science software is one of the keys to production planning and processing:

- Performs the actual data processing to create desired products.
- Is developed at Science Computing Facilities (SCFs) external to ECS.
- Is embodied in Product Generation Executives (PGEs) when the software is integrated into the ECS production processing environment.
    - PGEs are science software code (e.g., executable programs or shell scripts) that contain the instructions for processing data to create the desired products.

The production request (PR) is another key to production planning and processing. The Production Planner defines ECS science data processing in terms of PRs.

- A PR is an order for data to be produced by the Data Processing Subsystem.
- A single PR may specify several jobs (using the same PGE) that are to be run over a period of time or a single job producing a single set of data.
- PRs may apply to the processing of new data (standard PRs or standing orders) or the reprocessing of existing data (reprocessing PRs).
- Each PR identifies a specific PGE for generating a particular type of product.
    - Some PGEs are dependent on others; i.e., some PGEs require input data that are the output of other PGEs.
    - The planning software will recognize and reject a PR when the PR specifies a PGE that requires data from another PGE that has not yet been specified in a PR.

The Planning Subsystem performs the following functions:

- Uses each PR to generate either one or a series of Data Processing Requests (DPRs).
    - Each DPR corresponds to one execution of a single PGE.

— Each DPR contains the information that is needed by the SDP processing function, including PGE-related information.

- Checks the availability of the data required for the DPR, either from the data server (if the data have been previously ingested) or from internal predictions (if the data are expected to arrive in the future).
- Determines what data will be included in the DPR output so the system can make predictions concerning the availability of data for subsequent PGEs.

Figure 11.5.1-1 shows the relationships among the PGEs, PRs, and DPRs as they are accessed through the Production Request Editor GUI.

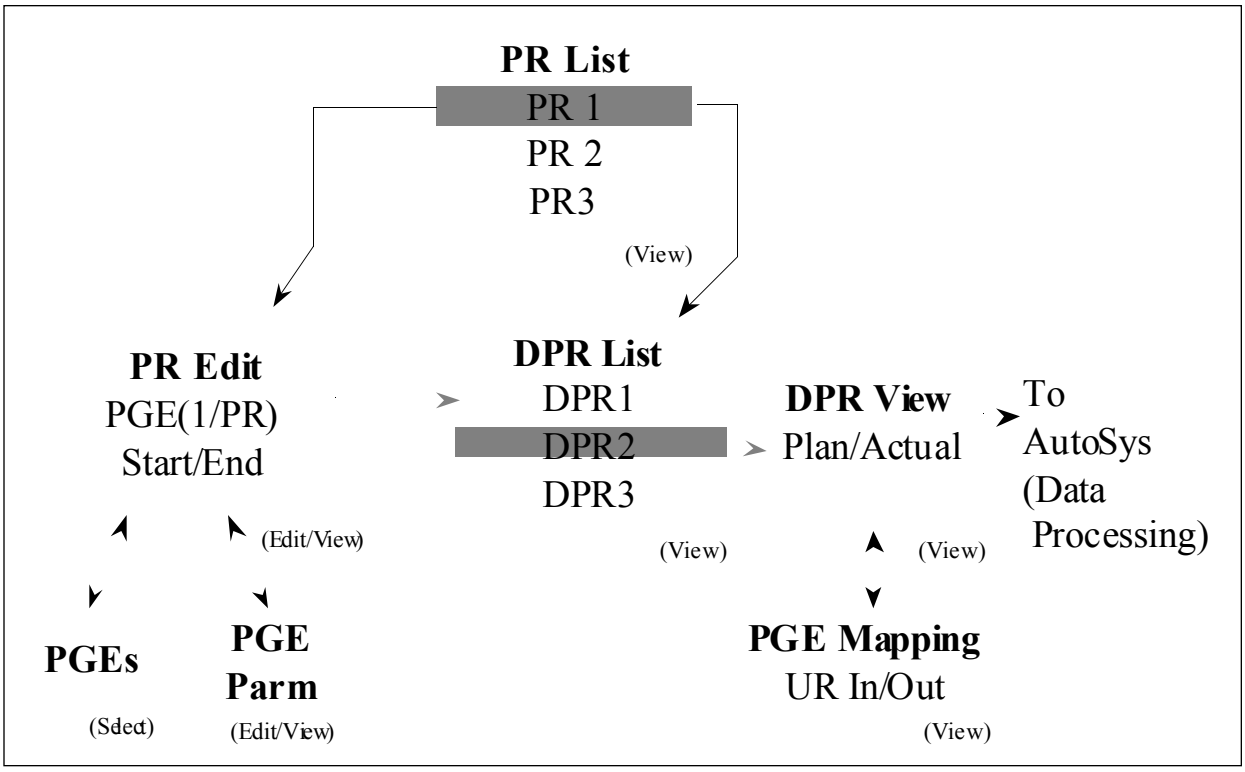

Figure 11.5.1-1. Production Request Editor Flow

11.5.2 Types of Processing

ECS either accommodates or will accommodate the following three general types of data processing:

- Routine Processing
- Reprocessing
- Ad-Hoc Reprocessing
- On-Demand Processing

11.5.2.1 Routine Processing

Routine Processing is pre-defined software production processing that is periodic and keyed to data arrival. For example, every day a Production Planner includes in the daily schedule a DPR for generating a particular Level 1A product from the most recent Level 0 data from the applicable satellite instrument.

11.5.2.2 Reprocessing

Reprocessing typically involves using a new, improved PGE to process data that had previously been processed with an older version of the PGE. In such cases reprocessing would be a large-scale operation, especially if several years worth of data were to be reprocessed. Consequently, the Production Planner is likely to schedule reprocessing in manageable quantities so the processing resources can accommodate routine and on-demand processing in addition to the reprocessing.

11.5.2.3 Ad-hoc Reprocessing

Ad-hoc Reprocessing could be necessary at any time. For example, if a product fails a quality assurance (QA) check, the same PGE could be run again on the same data set in the hope of creating an acceptable product. Similarly, if processing of a PGE fails for some reason, it might be possible to rerun the PGE and hopefully achieve a successful outcome.

11.5.2.4 On-Demand Processing

On=Demand Processing is ad-hoc processing initiated by either the Planning Subsystem or an end-user (as opposed to the Production Planner). For example, a researcher using data from the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) instrument on the Terra satellite may need a particular Level 2 product that has not yet been generated. The ASTER researcher would submit an on-demand request to have the product generated from a Level 1B product stored in the archive.
In the future such on-demand processing requests (OPRs) will be entered from a Client Subsystem tool, passed through the Distributed Information Manager (Data Management Subsystem) and the Data Server to the Planning Subsystem. Currently there is a work-around to the automated process which requires the requester to contact DAAC personnel to make the request. So far ASTER researchers are the only identified external users of automated on-demand processing. Another future feature is automated cross-DAAC planning. It is a process that will be undertaken when products produced at one DAAC require inputs being produced at another DAAC. The predicted production time of remote input products will be used in creating local production plans. The primary mechanism for cross-DAAC planning will be the use of Predicted Data Availability Schedules (PDAS), created when a plan is created. A DAAC's PDAS will be made available to remote DAACs via the Data Server.

11.5.2.5 ASTER on Demand Capability

There is some information that needs to be provided to the operations personnel who will be doing the SSIT procedures that create the many PGE Profiles that the ASTER on Demand capability depends on. When ODFRM passes ODMGR the parameters that a user has set for a particular Product they come in a parameter = value format in the GlParameterList. What ODMGR does is attempt to match these parameter/value pairs against Runtime Parameter/Value pairs in the PDPS database that have been entered at SSIT. If the parameter or values do not EXACTLY match what was put in the ODL for the Runtime parameter/value pairs, then ODMGR will not be able to select the correct PGE to create the product.

11.5.2.6 AST_06 Product Guidelines

The are certain parameters that are required for the AST_06 products. However, not all of the parameters have been identified yet.We needed to have 3 different PGEs, each of which produces a different AST_06 product (AST_06VD, AST_06SD, AST_06TD).
For the AST_06 products, the parameters should NOT be set (in the ODL) as selector parameters. The parameters for the AST_06 products are what are called "overwriteable" parameters, meaning that PDPS will overwrite the value in the database with whatever it gets from ODFRM. The parameters that can be set when requesting the AST_06 products are as follows:

MatrixType
SamplingFreq
OutputMean
OutputSigma
BlueBand
GreenBand
RedBand
FirstLine
LastLine
FirstSample
LastSample\

11.5.2.7 Non-Standard L1B on Demand Requests

In support of Non-Standard L1B On Demand Requests, SSI&T personnel at EDC need to add information to the ODL defining the AST_L1B ESDT to PDPS. The installation can set up the database originally but any re-registration of the AST_L1B ESDT (in the PDPS database) will cause the information populated in the database to be deleted.
NCR 25773 has been written against this problem, but unknown is the timeframe for it being fixed. So the ODL file ESDT_AST_L1B#{ESDT version}.odl needs to have the following added for Non-Standard L1B requests:

```
OBJECT = METADATA_DEFINITION
      CLASS = 1
      PARM_NAME = "InputPointer"
      CONTAINER_NAME = "InputGranule"
      TYPE = "STR"
    END_OBJECT = METADATA_DEFINITION

    OBJECT = METADATA_DEFINITION
      CLASS = 2
      PARM_NAME = "ASTERMapProjection"
      CONTAINER_NAME = "AdditionalAttributes"
      TYPE = "STR"
    END_OBJECT = METADATA_DEFINITION

    OBJECT = METADATA_DEFINITION
      CLASS = 3
```

```
    PARM_NAME = "Resampling"
    CONTAINER_NAME = "AdditionalAttributes"
    TYPE = "STR"
  END_OBJECT = METADATA_DEFINITION
```

11.6 Production Rules used in Planning PGE Processing

11.6.1 Production Rules Defining the PGE

Production rules are the instructions about how a particular PGE is to be run. The instructions specify a wide range of information such as the input and output data types, the frequency of execution, activation conditions and error handling instructions.

A single PGE may use one or more sets of production rules, known as PGE profiles, since it may be desirable to run the same PGE with different input data sets, or activation conditions. The production rules are entered when a PGE undergoes Science Software Integration and Test (SSI&T) at the DAAC. Where applicable, default parameter values are entered at that time. The initially selected runtime parameters, metadata check parameters, tile IDs, and many of the default parameters can be overridden in the production environment when a Production Request is entered.

Production rules define the PGE to the Planning and Data Processing Subsystems (PDPS). The following types of conditions can be specified for each PGE:

- The time period for which the PGE will run.
  - A PGE can run every hour, every day, or for every orbit of a satellite. The frequency of how often a PGE runs must be defined to PDPS so that it knows when to plan and execute the PGE. A definition of a satellite's orbit could be included if the PGE were to be executed for some number of satellite passes.
- PGE Inputs.
  - A PGE can have any number of inputs. The types of the inputs and how frequently they are available helps determine on what basis the PGE is scheduled.
  - Most inputs to a PGE are retrieved based on time; the specified inputs are retrieved from the Data Server Subsystem for the time which the PGE is defined to execute. Production Rules allow other conditions to be added to the mix, such as checks or queries against the metadata of the input granules, or the lists of inputs as alternates (for when a primary input is not available) or optionals (for inputs without which the PGE can still run successfully). If inputs are defined as alternate or optional, the number of inputs staged for the execution of the PGE may vary from one run to the next.
- PGE Outputs.

- A PGE can have any number of outputs. The characteristics of the outputs can have effects on any downstream PGEs that use them as inputs. For example, it is possible for an output to be defined as optional, in which case it may or may not even be produced. (When an output is not generated, it cannot be used as input for a downstream PGE.)

- Runtime Parameter Values.
  - A PGE can have any number of runtime parameters, which are values that are placed in the process control file (PCF) under specified logical IDs before the PGE executes. The PGE treats them as constants and normally they are set either during SSI&T or when the Production Request is entered.
  - For some production rules (such as Orbital Processing) there is specific information that can be placed in a runtime parameter if so desired by the PGE.
- Geographic Tiles.
  - A PGE can define a geographic location for which it will process data. Tiles are defined through production rules, and change the staging of inputs from time-based, to a combination of time- and geographically-based. Data are retrieved based on their location on the Earth with respect to the tile that it is currently being processed.

Some (but not all) production rules can work with other production rules.

11.6.1.1 Production rules are often used for the selection of dynamic inputs.

- **Dynamic inputs** can be either internal or external.
  - **Dynamic internal** inputs are produced by other PGEs (they are called dynamic internal inputs because they are produced at an ECS DAAC).
  - **Dynamic external** inputs are periodically ingested and stored in the Data Server Subsystem (they are termed dynamic external inputs because they are produced outside of the DAAC).
- **Static inputs** are granules that are inserted during the SSI&T process and are retrieved not on the basis of time but by Earth Science Data Type (ESDT) and science group.
  - The Metadata Query Production Rule is the only production rule that works for choosing static inputs.

PGE profiles allow a PGE to be defined to PDPS multiple times, each with a different set of inputs, outputs, or even scheduling information. Each PGE's definition is made up of its name, its version and its profile number. Different PGE name/version pairs define different PGEs to PDPS. The addition of the profile allows for multiple definitions of a PGE name/version pair. There can be up to 99 profiles for each PGE.

11.6.1.2 Syntax of Production Rules

Production rules are defined in the following two ways:

- Through science metadata that is entered in various types of files during the SSI&T process.
- By entering parameter values when a Production Request is created to schedule the PGE.

During SSI&T, production rules are defined in files written in Object Description Language (ODL) in a parameter equals value format. There are three general categories of ODL files:

- PGE Science Metadata ODL Files.
- ESDT Science Metadata ODL Files.
- Production Rule-Specific Science Metadata ODL Files
  - Orbit Definition ODL Files.
  - Path Map Definition ODL Files.
  - Tile Definition ODL Files.

When a Production Request is created to schedule a PGE, it is necessary to enter certain information that is essential to implementing the production rules that affect the particular PGE. The information may concern the date and time-range

11.6.1.3 PGE Science Metadata ODL Files

The PGE science metadata ODL file defines a PGE (or at least the current plan for its operation) to PDPS. It specifies everything from the PGE name and version, to the period for the PGE (how often it runs), all inputs and outputs, any runtime parameters and any exit messages or dependencies. A template version of the PGE science metadata ODL file is created by the **SSIT Create ODL Template** program from a PCF from the PGE.

11.6.1.4 ESDT Science Metadata ODL Files

The ESDT science metadata ODL file defines a PGE input or output to PDPS. Each input and output of a PGE must have a corresponding ESDT science metadata ODL file defined. It describes everything that PDPS needs to know about the subject input or output file, from its name and version, to its period (how often data is collected), to where it is used and archived. Note that many PGEs can use the same input or output type, and thus can share the same ESDT science metadata ODL file.

Unlike the PGE science metadata ODL file, there is no tool for automatically generating a template ESDT science metadata ODL file. A template version exists under the data directory called ESDT_ODL.template. The template must be copied to a file that follows the naming convention *ESDTShortName#ESDTVersionID*.odl.

11.6.1.5 Production Rule-Specific Science Metadata ODL Files

The production rule-specific science metadata ODL files provide specific information to PDPS about production rules used by a PGE. They are needed only when the PGE is subject to one of the following conditions:

- Is executed on the basis of a satellite orbit.
- Needs to know the orbital path of a satellite.
- Requires data based on geographic tiling of the Earth.

Since not every PGE is based on orbits or tiles, not all PGEs require these files. The comments in the PGE_ODL.template describe when setting a specific parameter means that a production rule-specific science metadata ODL file needs to be created.

The production rule-specific science metadata ODL files are broken into three types, which are defined as follows:

- Orbit ODL File.
  - Defines the orbital period of the satellite from which the PGE's input data is created.
  - Defines when a given orbit starts, how long it lasts, and the number of the orbit.
  - PDPS uses the information in the orbit ODL file to extrapolate future orbits and is able to plan PGEs that are required to run every so many orbits of the satellite.
- Pathmap ODL File.
  - Defines the mapping between the cyclic 0-233 orbits that the satellite makes with the actual path number that the PGE requires.
  - PDPS computes the path number from the orbit number (specified in the orbit ODL file) by incrementing it until it reaches the 233 maximum, then resetting it to zero.
  - Many instruments expect the path number to be a fixed swath on the Earth, so it is not just incremented for each satellite pass.
  - The pathmap ODL file creates a mapping from the sequential path numbers to the path numbers expected by the PGEs.
- Tile ODL File.
  - Defines the coordinates of the tiles used by some instruments to specify geographic locations on the Earth.
  - The tile definitions are used by PDPS to schedule the PGE (one execution per tile) and to acquire the necessary data (using the geographic coordinates to acquire data for the tile being processed only).

Unlike the PGE science metadata ODL file, there is no tool to automatically generate a template production rule-specific science metadata ODL file. Because the files themselves tend to be small, this is not usually a problem. A template version of each kind of production rule-specific science metadata ODL file (e.g., ORBIT_ODL.template, TILE_ODL.template) exists in the /usr/ecs/<MODE>/CUSTOM/data directory on the AIT Workstation. The templates must be copied, named properly, and edited in order to create the appropriate production rule-specific science metadata ODL file.

11.7 Release 5 Production Rules

The following statements provide some simplified descriptions of production rules that are scheduled to be made available in Release 5:

- **Basic Temporal** - Temporal (time) range of inputs matches the temporal range of outputs.
- **Advanced Temporal** - Temporal range of inputs is offset from the expected temporal range of inputs and outputs.
- **Alternate Input** - PGE is run with different inputs based on the availability of various alternate input data sets.
- **Optional Input** - PGE is run with specified optional inputs if available; otherwise, PGE is run without them.
- **Minimum/Maximum Number of Granules** - Minimum number of input granules needed for full data coverage and maximum number of input granules to search for may be specified. Minimum and maximum number of outputs expected from the PGE may be specified.
- **Optional DPRs** – The only DPRs executed are those for which the non-routine key input data actually become available (i.e., are either produced in data processing or can be acquired from the archive).
- **Intermittent Activation** - Every n^{th} DPR is activated; all other DPRs are skipped.
- **Metadata Checks** - DPR is run only if input data's metadata value(s) meet(s) certain criteria.
- **Metadata Query** - Input granule selection is based on metadata value.
- **Data Day** - Input data selection is based on Data Day.
- **Spatial Query** - Input granule selection is based on the spatial coverage of another input (i.e., the key input).
- **Tiling** - Input data is chosen on the basis of Instrument Team-defined tiles (geographic areas).
- **Closest Granule** – DPR is generated if a required input granule within a particular time range (rather than an exact time) is available; otherwise, no DPR is generated. (Supersedes the Most Recent Granule Production Rule)
- **Orbital Processing** - Selection of input times is based on orbit information.

11.7.1 Basic Temporal Production Rule

The Basic Temporal Production Rule defines the timeframe for the PGE along with its input and output data. PGEs subject to the Basic Temporal Production Rule generally have the following characteristics in common:

- Typically scheduled to run using input data that become available periodically (every hour, every day, etc.).
- Use input data for a particular period of time.
- Produce output for a specified length of time.

The data the PGE takes in (its input) and the data it produces (its output) have the same period (or some subset of the same period) as the PGE.

- Example One:

  - A MODIS PGE processes data for five-minute intervals, producing Level 1B granules.
  - The PGE requires as input the specific five-minute Level 1A granule that is contemporaneous with (covers the same five-minute time period as) the Level 1B granule to be produced.
  - Using the Basic Temporal Production Rule, a five-minute Level 1A granule is staged as input to the PGE and a five-minute Level 1B granule is expected as output, both matching the timeframe for which the PGE is run.
- Example Two:
  - A CERES PGE processes data for 24-hour intervals, producing 24-hour Level 1A granules as output.
  - As input the PGE takes Level 0 data that is ingested every two hours.
  - Using the Basic Temporal Production Rule, twelve two-hour Level 0 granules are staged as input to the PGE and a 24-hour Level 1A granule is expected as output, matching the timeframe for which the PGE is run.

The fundamental elements used to define the Basic Temporal Production Rule are "period: and "boundary."

- **Period** is the length of time for which a PGE processes data or the length of time for which input and output data is collected.
  - A PGE that is subject to the Basic Temporal Production Rule only and that processes data in two-hour blocks, takes in data that relates to a particular two-hour interval and produces output data for that same two-hour period.
  - Data that has a period of 15 minutes was collected or produced for a 15-minute time period.
- **Boundary** is the starting point for the data or PGE. Depending on the characteristics of the data or PGE, the boundary may be the start of a minute or hour or day or week (etc.).
  - If a PGE's boundary is the start of the hour, it processes data that starts every hour and runs on data for the length of its period.
  - If data comes in every day, PDPS predicts that the data is going to be available at the start of the day and allows scheduling of PGEs that use the data as input accordingly.

Both the PGE itself and the input data have a boundary and period associated with them. That is how PDPS determines the frequency of processing for a Basic Temporal PGE and the time period for its inputs and outputs.
PDPS uses **period** and **boundary** in combination to plan the processing of each PGE, including determining its input requirements and anticipated output (which may be input to other PGEs). If a PGE has a period of one hour and a boundary of "start of day," it is scheduled every hour, beginning at midnight. If an input has a period of 15 minutes and boundary of "start of hour," PDPS predicts it every 15 minutes beginning on the hour.

Boundary offset is an addition to the Basic Temporal Production Rule that allows a PGE or data to start on an offset from a given boundary. For example, if a PGE would normally run every day but not start until two or three hours into the day (e.g., beginning at 3:00 a.m. instead of midnight), a boundary offset can be used to add three hours to the "start of day" boundary. This would mean the PGE would run on data that occurred three hours after the boundary.
Data with offset times refers to data where the start time is a few minutes off of the start time that PDPS expects. For example: if data is defined to PDPS as follows:

```
BOUNDARY = "START_OF_HOUR"
PERIOD   = "HOURS=1"
```

but the data actually starts at 1:05 and ends at 2:05, the data is said to have **offset times**. There is a flag in the production rule syntax that tells PDPS to shift granule time specifications to match the granules in the archive.
The **end-of-month anomaly** is an addition to the Basic Temporal Production Rule that allows a PGE or data to cover a specific number of days within a month. The month is broken into thirds. The first third is composed of the first 10 days of the month. The second third consists of days 11 through 20. And the last third varies in length depending on the total number of days in the month (i.e., for November it would have 10 days; for December it would have 11 days). A specific **boundary** and **period** allow a PGE or its data to be scheduled into thirds of a month.
Figure 11.7.1-1 provides an illustration of the Basic Temporal Production Rule. The PGE has a boundary of "start of day" and a period of one hour, so it is scheduled for every hour through the day. If a Production Request were entered for two full days of processing, a DPR would be created for the PGE to run every hour; i.e., 48 DPRs total. If a Production Request were created for a four-hour period in the middle of a single day (for example, from 12:00 noon to 4:00 p.m.), then four DPRs would be created, one for 12:00-1:00, one for 1:00-2:00, one for 2:00-3:00, and one for 3:00-4:00.

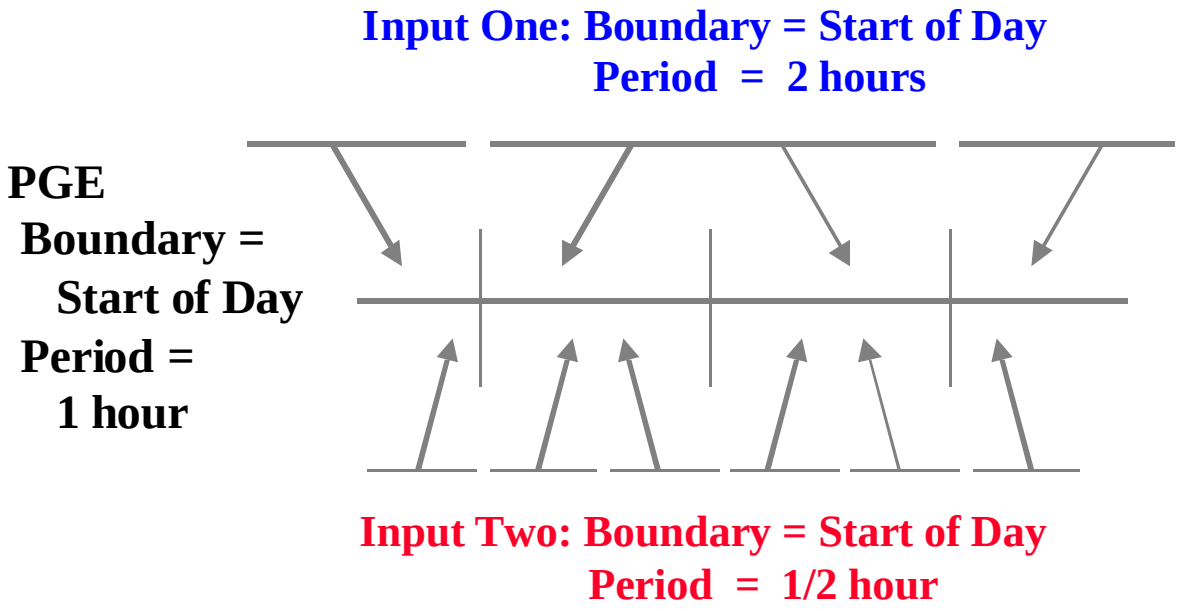

Figure 11.7.1-1. Example of the Basic Temporal Production Rule

In the example (Figure 11.7.1-1), Input One has a boundary of “start of day” and a period of two hours, so when PDPS plans for its availability, it expects a granule every two hours beginning at midnight. Consequently, each granule of Input One is associated with two DPRs for the PGE, because the PGE encompasses only one hour of the two-hour granule's period.
Input Two has a boundary of “start of day” and a period of _ hour, so when PDPS plans for its availability, it expects a granule every _ hour beginning at midnight. As a result two granules of Input Two are associated with each DPR for the PGE, because the PGE encompasses an hour of the _-hour granule's Period. Thus, every DPR of the PGE will wait for two granules of Input Two to arrive before it can be processed.

11.7.1.1 PGE Science Metadata ODL File Parameters

The following parameters must be set properly in the applicable PGE science metadata ODL file in order to implement the Basic Temporal Production Rule:

- SCHEDULE_TYPE.
- PROCESSING_PERIOD.
- PROCESSING_BOUNDARY.

The SCHEDULE_TYPE parameter specifies the type of scheduling that will be done for the PGE. The following values are applicable to the Basic Temporal Production Rule:

- "Time"
  - The PGE is scheduled on the basis of the specified boundary/period and the availability of data for that boundary/period.

- "Snapshot"
  - The PGE is scheduled for a single date/time.
  - Note that PROCESSING_PERIOD and PROCESSING_BOUNDARY are **not** needed when "Snapshot" is specified.

Other values for SCHEDULE_TYPE apply to other production rules, such as the following values:

- “Data"
  - The PGE is scheduled on the basis of the availability of data produced by other PGEs.
- "Tile"
  - The PGE is scheduled based on the definition of geographic tiles.
- "Orbit"
  - PGE scheduling is based on the orbit of the spacecraft.

The PROCESSING_PERIOD parameter describes the length of time for which the PGE executes. Data will be acquired (barring any combination of Production Rules) for the specified period and output data will be planned for the given period. It is of the format "<Period Type>=<Length of Period>". Note that “length of period” can be specified as a positive integer only. The following values are acceptable “period type” entries for the Basic Temporal Production Rule:

- "YEARS"
  - PGE processes data applicable to a given year or years.
  - "YEARS" might be specified for a PGE that computes a yearly average.
  - For example, PROCESSING_PERIOD = "YEARS=1" relates to a PGE that processes one year’s worth of data.
- "MONTHS"
  - PGE processes data applicable to a particular month or several months.
  - “MONTHS” is most likely to be used for some kind of averaging PGE.
  - For example, PROCESSING_PERIOD = "MONTHS=2" relates to a PGE that processes two months’ worth of data at a time.
- "THIRDS"
  - PGE processes data applicable to some number of thirds of the month.
  - For example, PROCESSING_PERIOD = "THIRDS=1" relates to a PGE that processes data applicable to 1/3 of the month.
- "WEEKS"
  - PGE processes data applicable to some number of weeks.

  - For example, PROCESSING_PERIOD = "WEEKS=2" relates to a PGE that processes two weeks' worth of data every time it runs.

- "DAYS"
  - PGE processes data applicable to some number of days.
  - For example, PROCESSING_PERIOD = "DAYS=5" relates to a PGE that processes five days' worth of data.
- "HOURS"
  - PGE processes data applicable to some number of hours.
  - For example, PROCESSING_PERIOD = "HOURS=4" relates to a PGE that processes four hours' worth of data when it is executed.
- "MINS"
  - PGE processes data applicable to some number of minutes.
  - For example, PROCESSING_PERIOD = "MINS=5" relates to a PGE that processes five minutes' worth of data.
- "SECS"
  - PGE processes data applicable to some number of seconds.
  - For example, PROCESSING_PERIOD = "SECS=2" relates to a PGE that runs on two seconds' worth of data.

There are other types of values for PROCESSING_PERIOD but they apply to other production rules (as described in the applicable sections of the lesson).

The PROCESSING_BOUNDARY parameter specifies the boundary (starting point in time) of the PGE. It tells when each instance of the PGE should start. Note that the PROCESSING_BOUNDARY and PROCESSING_PERIOD are used in conjunction to schedule the PGE.

The following PROCESSING_BOUNDARY values are used for implementing the Basic Temporal Production Rule:

- "START_OF_HOUR" – PGE processes data for each hourly interval.
- "START_OF_6HOUR" - PGE processes data for every 6-hour interval.
- "START_OF_DAY" - PGE processes data for every daily interval.
- "START_OF_WEEK" - PGE processes data for every weekly interval.
- ↑ "START_OF_ONE_THIRD_MONTH" - PGE processes data for every 1/3 of a month.
- "START_OF_MONTH" - PGE processes data for every monthly interval.
- "START_OF_YEAR" - PGE processes data for every yearly interval.
- "START_DATE=*DD/MM/YYYY*" - PGE processes data for the specified date only.

There are other values for PROCESSING_BOUNDARY that apply to other production rules (as described in the applicable sections of the lesson).

11.7.1.2 Handling Data with Offset Times

When the ALIGN_DPR_TIME_WITH_INPUT_TIME flag is set to "Y" (i.e., ALIGN_DPR_TIME_WITH_INPUT_TIME = “Y”) PDPS shifts the expected times for input data to the actual times found in the archive. If the flag is NOT set, data with offset times can cause problems when generating Production Requests.

11.7.1.3 ESDT Science Metadata ODL File Parameters

The following parameters must be set properly in the applicable ESDT science metadata ODL file in order to implement the Basic Temporal Production Rule:

- DYNAMIC_FLAG.
- PERIOD.
- BOUNDARY.

The DYNAMIC_FLAG describes the type of data that is defined in the ESDT science metadata ODL file. It specifies to PDPS what kind of data the PGE requires as input or produces as output. It can have any of the following four possible values, all of which are valid for Basic Temporal data:

- "S"
  - Static Data.
  - Data do not change at regular intervals.
  - The same granule can be used as input for many runs of the PGE.
  - Calibration files are a good example of static data.
- "I"
  - Dynamic Internal.
  - Data are produced by a PGE running at the local DAAC.
  - All output products are either “dynamic internal” or “interim” kinds of data.
- "E"
  - Dynamic External.
  - Data are produced by an external source (not a PGE running at the local DAAC).
  - EDOS data is a primary example.
  - Dynamic external can be set for PGE inputs only.
- "T"
  - Interim/Intermediate.
  - Data are stored only temporarily by the Data Server Subsystem.

The PERIOD parameter specifies the length of time covered by the data. Data are expected to be either ingested or produced for the length of the PROCESSING_PERIOD described in PGE science metadata ODL files. However, the PERIOD of the data does **not** have to match the PROCESSING_PERIOD defined for the PGE. PDPS plans for data where the ESDT period is

less or more than the processing period of the PGE that uses it. For example, if the PGE PROCESSING_PERIOD = "HOURS=1" and the input data PERIOD = "MINS=5", then PDPS plans to acquire twelve granules of the input data to cover the PROCESSING_PERIOD.
The following “period type” values are used for implementing the Basic Temporal Production Rule:

- "YEARS"
  - Data span a year or years.
  - “YEARS” might be selected for a yearly average output product.
  - For example, PERIOD = "YEARS=1" specifies data that cover a period of a year.
- "MONTHS"
  - Data span a month or several months.
  - “MONTHS” is most likely used for some kind of averaging output product.
  - For example, PERIOD = "MONTHS=2" specifies data that cover a period of two months.
- "THIRDS"
  - Data span some number of thirds of a month.
  - For example, PERIOD = "THIRDS=1" specifies data that cover a period of 1/3 month.
- "WEEKS"
  - Data span some number of weeks.
  - For example, PERIOD = "WEEKS=2" specifies data that cover a period of two weeks.
- "DAYS"
  - Data span some number of days.
  - For example, PERIOD = "DAYS=5" specifies data that cover a period of five days.
- "HOURS"
  - Data span some number of hours.
  - For example, PERIOD = "HOURS=4" specifies data that cover a period of four hours.
- "MINS"
  - Data span some number of minutes.
  - For example, PERIOD = "MINS=5" specifies data that cover a period of five minutes.

- "SECS"
  - Data span some number of seconds.
  - For example, PERIOD = "SECS=2" specifies data that cover a period of two seconds.
- "ORBITS"
  - Data span some number of orbits of the spacecraft.
  - For example, PERIOD = "ORBITS=1" specifies data that cover one orbit.
  - A PGE can be time-scheduled (using the Basic Temporal Production Rule) but use orbit-based data.

The BOUNDARY parameter is the starting point in time of the data granule. It tells when each data granule should start. Note that the BOUNDARY and PERIOD are used in conjunction to determine the starting and ending time for the granules.
The following values for BOUNDARY apply to the Basic Temporal Production Rule:

- "START_OF_HOUR"
  - Data granules start every hour.
- "START_OF_6HOUR"
  - Data granules start every six hours.
- "START_OF_DAY"
  - Data granules start every day.
- "START_OF_WEEK"
  - Data granules start every week.
- "START_OF_ONE_THIRD_MONTH"
  - Data granules start every 1/3 of a month.
- "START_OF_MONTH"
  - Data granules start every month.
- "START_OF_YEAR"
  - Data granules start every year.
- "START_OF_ORBIT"
  - Data granules start every orbit.

11.7.2 Advanced Temporal Production Rule

The Advanced Temporal Production Rule allows for input data to be acquired for a time period other than that of the PGE or its planned inputs/outputs. It provides an offset mechanism, specifying on an input basis that the data required for processing is some number of seconds earlier or later than the planned time period for the PGE.

- Example One:

- A PGE requires data from its previous execution for interpolation purposes (e.g., one of its inputs is the output of the very same PGE the last time that it ran).
- If the PGE processes data for each one-hour interval (producing an hourly product), the Advanced Temporal Production Rule is specified with an offset of minus 3600 seconds (one hour) for the input of the ESDT produced by previous runs.

- Example Two:
  - A PGE takes as input two-hour Level 0 data to produce an L1A product.
  - Because the edges of the Level 0 data can be difficult to process without preceding and succeeding data, the PGE requires three Level 0 granules, one from the time period before it runs, one for the time period it is currently processing and one for the next time period.
  - The PGE is defined as having three inputs, the first with and Advanced Temporal offset of minus 7200 seconds (two hours), the second with no Advanced Temporal offset and the third with an Advanced Temporal offset of plus 7200 seconds (two hours).

The Advanced Temporal Production Rule uses the times specified in the Basic Temporal Production Rule as a reference point for specifying offset(s) to request data from a "period" and/or "boundary" different from that of the DPR or its input. The offsets are specified as either negative or positive numbers to indicate whether the time period of the input data is before or after that of the DPR (a particular run of a PGE).

- ↑ **Begin Period Offset** is an amount of time (in seconds) that is specified with respect to the DPR start time. A negative beginning offset requests data that was collected before the DPR start time. A positive beginning offset requests data with a collection time after the start time of the DPR.
- ↑ **End Period Offset** is an amount of time (in seconds) that is specified with respect to the DPR end time. A negative ending offset requests data that ended collection before the DPR end time was reached. A positive ending offset requests data that ended collection after the end time of the DPR boundaries.

Note that the beginning and ending offsets are not absolute cut-offs for data. Overlapping granules (granules that start or end outside of the offsets) will be staged as inputs to the DPR. Figure 11.7.2-1 provides an illustration of the Advanced Temporal Production Rule. The PGE shown in the example processes data for every one-hour interval. However, Input One comes in at two-hour intervals and Input Two is produced every 1/2 hour.

Both the Begin Period Offset and End Period Offset for Input One are -7200 seconds (minus two hours). Consequently, every DPR will stage the "previous" Input One. This could be used to get the "previous" or "next" granule of an input.

The Begin Period Offset for Input Two is zero, meaning that it will match the Start Time of the DPR. The End Period Offset is +1800 seconds (plus 1/2 hour). Therefore, all Input Two granules will be staged that fall within the time period of the DPR plus 1/2 hour. The effect is to acquire all Input Two granules within the time period of the DPR, plus the one from the next 1/2-hour time period, for a total of three granules. The additional granule acquired by means of the End Period Offset might be used for interpolation purposes at the end point.

The same types of parameter settings that apply to the Basic Temporal Production Rule apply to the Advanced Temporal Production Rule. In addition, there are some parameters in the PGE science metadata ODL file that apply specifically to the Advanced Temporal Production Rule. However, the values applicable to the Basic Temporal Production Rule must be set before the Advanced Temporal Production Rule syntax is added.

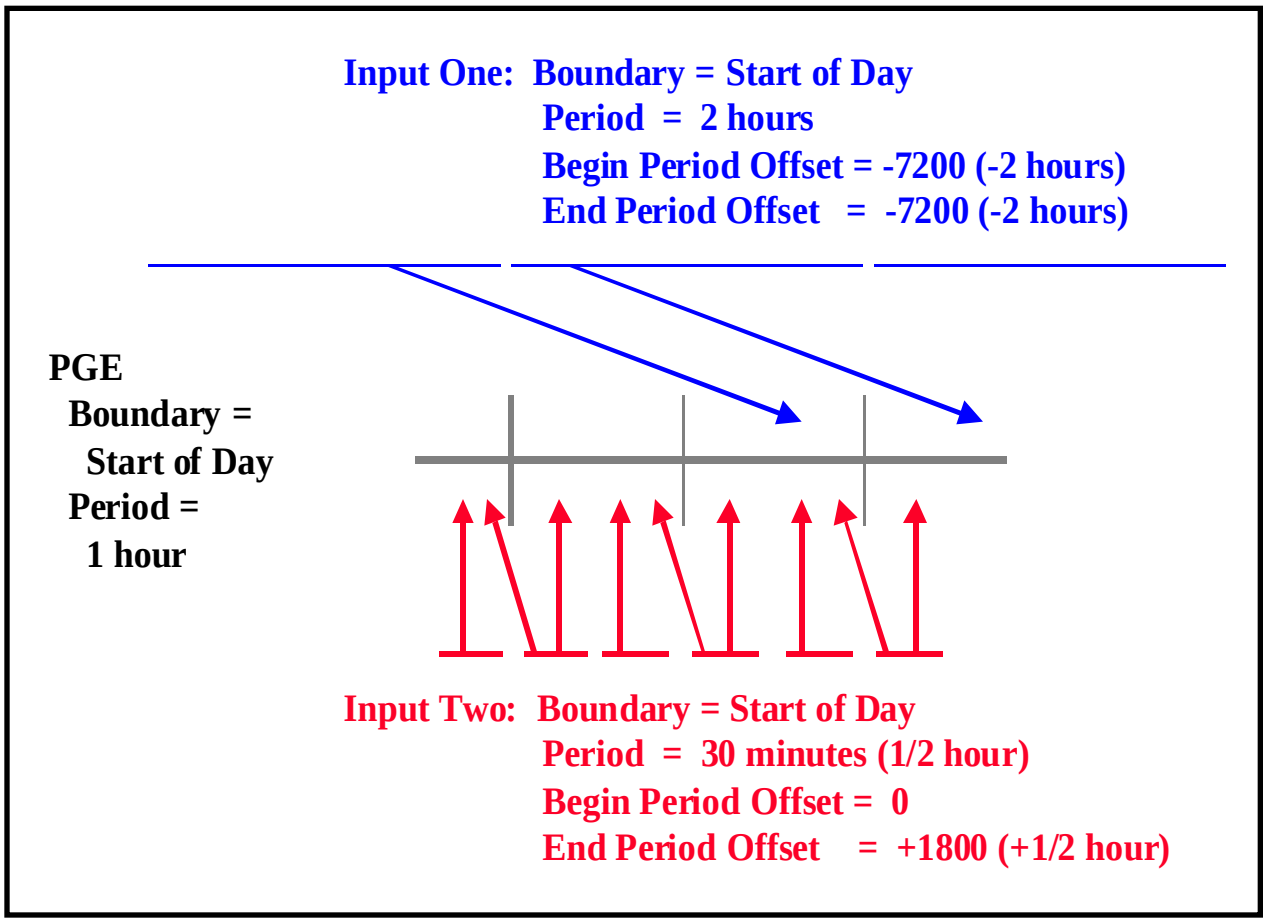

Figure 11.7.2-1. Example of the Advanced Temporal Production Rule

11.7.2.1 PGE Science Metadata ODL File Parameters

During the SSI&T process the PGE science metadata ODL file is generated from the PCF delivered with the science algorithm. A PCF_ENTRY object is generated for each file entry in the PCF. In order to implement the Advanced Temporal Production Rule the PCF_ENTRY object for each type of input file to which the rule applies uses the following syntax:

```
OBJECT = PCF_ENTRY
   .
   .
   .
   BEGIN_PERIOD_OFFSET =
```

```
        END_PERIOD_OFFSET =
        .
        .
        .
    END_OBJECT = PCF_ENTRY
```

Accordingly, the following parameters must be set properly in order to implement the Advanced Temporal Production Rule:

- BEGIN_PERIOD_OFFSET.
- END_PERIOD_OFFSET.

BEGIN_PERIOD_OFFSET is the offset added to or subtracted from the Data Start Time of the DPR. The value assigned to BEGIN_PERIOD_OFFSET can be either a positive or negative value, specified in seconds. If the value is positive, it is added to the Data Collection Start Time (looking for the input forward in time). If the value is negative, it is subtracted from the Data Collection Start Time (looking backward in time). For example, BEGIN_PERIOD_OFFSET = -3600 requests data that was collected one hour (3600 seconds) before the DPR start time.
END_PERIOD_OFFSET is the offset added to or subtracted from the Data Collection End Time of the DPR. The value assigned to END_PERIOD_OFFSET can be either a positive or negative value, specified in seconds. If the value is positive, it is added to the Data Collection End Time (looking for the input forward in time). If the value is negative, it is subtracted from the Data Collection End Time (looking backward in time). For example, END_PERIOD_OFFSET = +2700 requests data that was collected 45 minutes (2700 seconds) after the DPR end time.
The BEGIN_PERIOD_OFFSET and END_PERIOD_OFFSET parameters can be specified for any input PCF_ENTRY in the PGE science metadata ODL file. If not specified, the parameters are set to zero (0) and the Advanced Temporal Production Rule does not apply to the PGE.

11.7.3 Alternate Input and Optional Input Production Rules

The Alternate Input and Optional Input Production Rules are very similar and use much the same processing in PDPS. Both rules allow a PGE to select various inputs based on timers and priority lists. The major difference is that Alternate Inputs requires that one of alternates on the list be used, whereas Optional Inputs allows successful execution of the PGE if no optional input on the list is available.
The Alternate Input Production Rule allows for a PGE to evaluate a list of inputs in priority order and be scheduled and executed with the best priority input that could be found. In essence, a PGE using Alternate Inputs is saying "I would like to run with Input A, but if it's not available, I am willing to run with Input B." A timer can be used to specify how long to wait for a given alternate choice before proceeding with a choice of lesser priority. The PGE is not executed until one of the alternate choices has been found.

- Example:
  - A PGE requires model wind data as an input but is capable of accepting wind data from a Data Assimilation Office (DAO) model, a National Centers for Environmental Prediction (NCEP) model, or (as a last resort) climatology.
  - The PGE would use the Alternate Input Production Rule to list each input in priority order, giving a timer value for how long to wait before trying the next input.
  - If the DAO data are most desirable, DAO would be listed as first choice or "primary" data.

- NCEP would be the second choice.
- Climatology would be the last choice.
- If a timer value is specified for DAO data, the PGE will wait for that timer to expire before running with either NCEP data or climatology.
- If a timer had been placed on the NCEP input, the PGE would wait before running with the climatology data.

11.7.3.1 The Optional Input Production Rule

Allows for a PGE to list inputs that are desired but not required for it to execute. The inputs are ranked as previously stated and timers are set to wait before choosing a lower-priority type of input. However, if none of the inputs on the list becomes available, the PGE starts because the alternatives are classified as "optional." In essence the PGE is saying "I would like to run with Input A, but if its not available, I can run (and produce reasonable output) without it."

- Example:
  - It would be preferable to run a particular MODIS PGE with the output of a MISR PGE as input.
  - However, the MISR output may not be produced every day.
  - So the MODIS PGE lists the MISR input as optional with a two-hour timer.
  - On those occasions when no MISR output is produced, the MODIS PGE waits for two hours and then is executed without the MISR input.

Figure 11.7.3-1 provides an illustration of the Alternate Input Production Rule. The PGE in the illustration has two inputs that are “required” so they must available for the PGE to be run. It also has one input that is “alternate.” The alternate input can be one of three choices, the first choice is the **primary**, then there are second and third choices.
After the pair of required inputs has become available, the alternate inputs are evaluated as follows:

- ↑ If the primary alternate is available, it is used as input and the PGE is scheduled for execution.
- ↑ There is a one-hour timer on the primary alternate. If the primary alternate is unavailable, the PGE waits until the primary alternate becomes available or the one-hour timer expires, whichever occurs first.
- ↑ If the second alternate is available after the timer for the primary alternate has expired, the second alternate is used as input and the PGE is scheduled for execution.

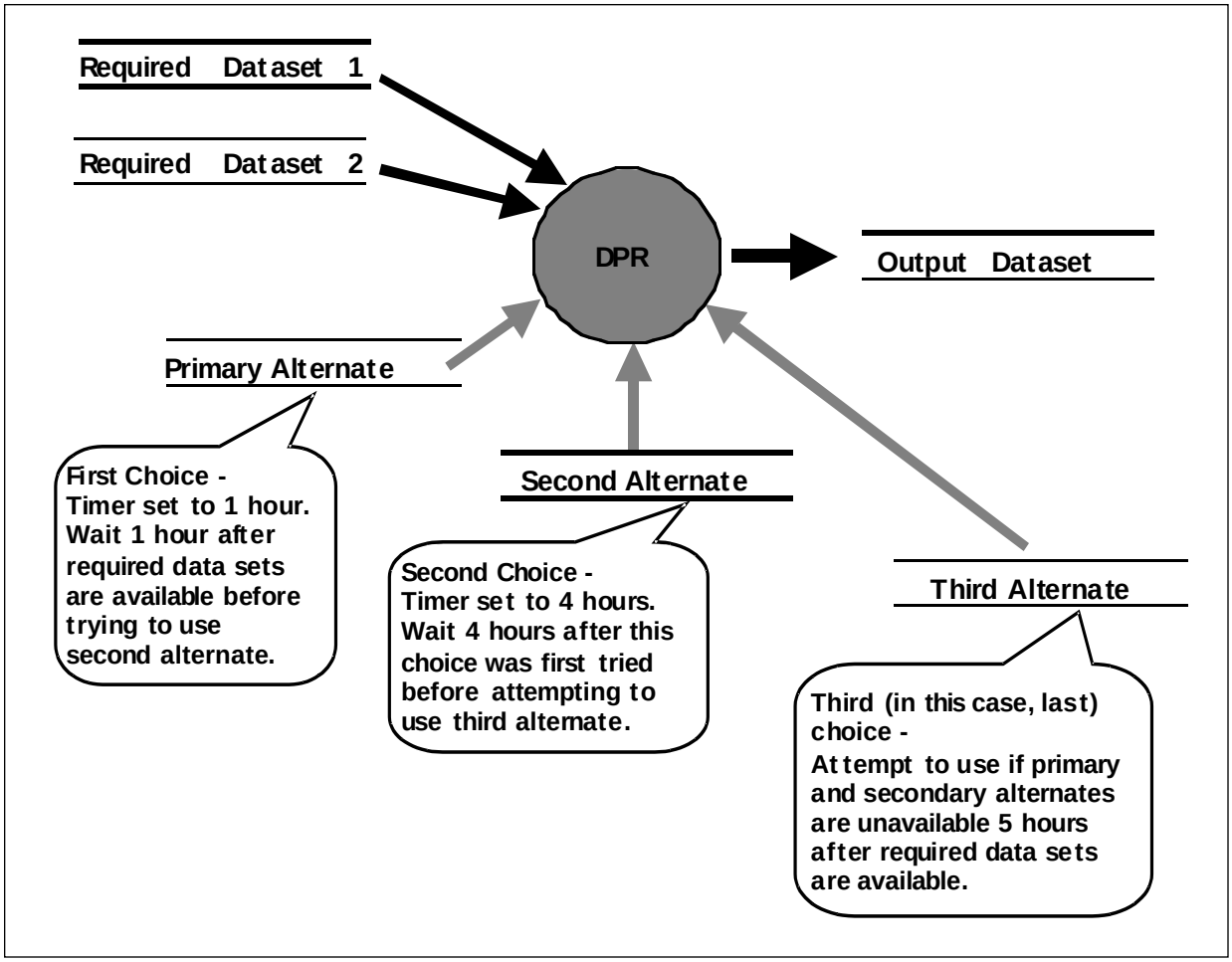

Figure 11.7.3-1. Example of the Alternate Input Production Rule

- ↑ There is a four-hour timer on the second alternate. If the second alternate is unavailable, the PGE waits until either the primary alternate or the secondary alternate becomes available or the four-hour timer expires, whichever occurs first.
- ↑ If the third alternate is available after the timer for the second alternate has expired, the third alternate is used and the PGE is scheduled for execution.
- ↑ There is no timer on the third alternate. If the third alternate is not available, the PGE waits until either the primary alternate, the secondary alternate, or the third alternate becomes available, whichever occurs first.
- The PGE will not start processing until one of the alternates becomes available.

If instead of an alternate the third input for the PGE had been defined as an optional input, the preceding scenario would have been the same, except that if neither the primary alternate, the second alternate nor the third option was available after the timers had expired, the PGE would not wait; it would be scheduled for execution without the third input. It would run with the two required inputs only.

The Alternate Input and Optional Input Production Rules are additions to settings/syntax put into the ODL files for other production rules. Inputs deemed “optional” or “alternate” can be

searched for and acquired by other production rules (e.g., Basic Temporal or Metadata Checks/Query). The syntax for the rules used to search for the inputs have to be filled out in addition to the syntax required to make the input an alternate or optional input.

11.7.3.2 PGE Science Metadata ODL File Parameters

The following parameter must be set properly in the applicable PGE science metadata ODL file in order to implement the Alternate Input or Optional Input Production Rule:

- INPUT_TYPE.

In addition, one of the following two ODL objects is used within a PCF_ENTRY to define either the Alternate Input Production Rule or the Optional Input Production Rule:

- ALTERNATE_INPUT object.
- OPTIONAL_INPUT object.

INPUT_TYPE is a type of data defined by a PCF_ENTRY object (i.e., between OBJECT = PCF_ENTRY and END_OBJECT = PCF_ENTRY). It can have one of four possible values, only three of which are used to define an alternate or optional inputs:

- "Required"
  - A required input.
  - The data must be available or the PGE does not execute.
  - It is the "normal" value for the parameter (i.e., INPUT_TYPE = “Required”); consequently, the input is neither an alternate input nor an optional input.
- "Primary"
  - The primary alternate input.
  - The data is the first choice in a list of alternates.
- "Alternate"
  - An alternate input (except the primary alternate) in a list of alternates.
  - The data is not the first choice in a list of alternates; it is a subsequent choice if the primary (or a higher-priority alternate) is not available.
- "Optional"
  - An optional input.
  - Availability of the data will be checked and if a timer has been specified, execution of the PGE will wait.
  - The PGE can be executed without the data if it is not available.

Although the Alternate Input and Optional Input Production Rules are similar, there are two different ODL objects used to define them within a PCF_ENTRY; i.e., the ALTERNATE_INPUT object and the OPTIONAL_INPUT object.
The ALTERNATE_INPUT object has the following syntax:

```
OBJECT = PCF_ENTRY
  .
  .
```

```
 .
.
.
OBJECT = ALTERNATE_INPUT
  .
  .
  .
 END_OBJECT = ALTERNATE_INPUT
END_OBJECT = PCF_ENTRY
```

The ALTERNATE_INPUT ODL object surrounds an Alternate Input definition. An OBJECT/END_OBJECT pair separates the parameters defining the Alternate Input from the rest of the parameters defining the PCF_ENTRY. The following parameters define an ALTERNATE_INPUT object:

- CLASS.
- CATEGORY.
- ORDER.
- RUNTIME_PARM_ID.
- TIMER.
- WAITFOR.
- TEMPORAL [not implemented].

CLASS is a simple counter used to differentiate the different ALTERNATE_INPUT objects within the file. Since each ALTERNATE_INPUT object resides within a different PCF_ENTRY object, the CLASS for an ALTERNATE_INPUT object can always be 1.
CATEGORY is the name of the list of alternates to which the ALTERNATE_INPUT belongs. The PDPS uses CATEGORY to associate different alternates within a list. CATEGORY can be set to any string value of 20 characters or less (e.g., CATEGORY = "Snow Ice"). Alternates that are part of the same list should have matching CATEGORY values.
ORDER is the numerical place that the particular alternate holds in the list of alternates. The first choice or Primary Alternate (with the INPUT_TYPE = "Primary") should have ORDER = 1.
RUNTIME_PARM_ID specifies the Logical ID (in the PCF) for which the PGE will find the Logical ID of the alternate chosen. Since all alternates must be contained within different PCF_ENTRY objects, they all must have different Logical IDs (but all alternates within the same CATEGORY should have the same value of RUNTIME_PARM_ID). The RUNTIME_PARM_ID parameter specifies the Logical ID of a runtime parameter that the PGE may read to find out which alternate was chosen for the particular execution of the PGE.
The TIMER parameter specifies how long to wait for the particular alternate before checking for the next alternate in the list. The parameter value is expressed in the format "<Period Type>=<Length of Period>". Note that “Length of Period” can be specified as a positive integer only. The Alternate Input Production Rule accepts the following “Period Type” values:

- "WEEKS"
  - PDPS should wait for some number of weeks before searching for the next alternate in the list.
  - For example, TIMER = "WEEKS=2" would make PDPS wait two weeks before checking for the next alternate input.
- "DAYS"

  - PDPS should wait for some number of days before searching for the next alternate in the list.
  - For example, TIMER = "DAYS=5" would make PDPS wait five days before checking for the next alternate input.
- "HOURS"
  - PDPS should wait for some number of hours before searching for the next alternate in the list.
  - For example, TIMER ="HOURS=4" would make PDPS wait four hours before checking for the next alternate input.
- "MINS"
  - PDPS should wait for some number of minutes before searching for the next alternate in the list.
  - For example, TIMER = "MINS=5" would make PDPS wait five minutes before checking for the next alternate input.
- "SECS"
  - PDPS should wait for some number of seconds before searching for the next alternate in the list.
  - For example, TIMER = "SECS=2" would make PDPS wait two seconds before checking for the next alternate input.

The WAITFOR parameter specifies whether or not the PGE can be run without the alternate input. Setting WAITFOR = "N" means that the PGE can run without the input if it cannot be found. In a list of alternate inputs, this would have meaning for the last choice only. If WAITFOR = "Y", the PGE is not executed (even after the last alternate timer expires) until one of the alternates in the list can be found.
The TEMPORAL parameter is an unimplemented feature that would allow for searching for alternates from the same time period but a different date. It is currently stored in the PDPS database but is not used.
The OPTIONAL_INPUT object has the following syntax:

```
OBJECT = PCF_ENTRY
  .
  .
  .
  .
  .
  OBJECT = OPTIONAL_INPUT
    .
    .
    .
  END_OBJECT = OPTIONAL_INPUT
END_OBJECT = PCF_ENTRY
```

The OPTIONAL_INPUT ODL object surrounds an Optional Input definition. An OBJECT/END_OBJECT pair separates the parameters defining the Optional Input from the rest of the parameters defining the PCF_ENTRY. The following parameters define an OPTIONAL_INPUT object:

- CLASS.
- CATEGORY.
- ORDER.
- RUNTIME_PARM_ID.
- TIMER.
- TEMPORAL [not implemented].

The parameters that apply to the Optional Input Production Rule are defined in the same way that the corresponding parameters are defined for the Alternate Input Production Rule. However, note that the Optional Input Production Rule has no WAITFOR parameter. It is irrelevant; in fact, the very essence of the Optional Input Production Rule depends on not "waiting for" the last option but going ahead with the execution of the PGE without the unavailable optional input(s).

Table 11.7.3.2-1. Extract of PGE Metadata ODL File Template Showing Alternate Inputs

```
>OBJECT = PCF_ENTRY
>   CLASS = 16
>   LOGICAL_ID = 1500
>   PCF_FILE_TYPE = 1
>   DATA_TYPE_NAME = "MOD10L2G"      [MODIS Level 2G Snow Cover]
>   DATA_TYPE_VERSION = "1"          [ESDT versioning in release B.0]
>   DATA_TYPE_REQUIREMENT = 1
>   SCIENCE_GROUP = ""
>   OBJECT = FILETYPE
>      CLASS = 1
>      FILETYPE_NAME = "Single File Granule"
>   END_OBJECT = FILETYPE
>   INPUT_TYPE = "Primary"
>   NUMBER_NEEDED = 1
>   OBJECT = ALTERNATE_INPUT
>      CLASS = 1
>      CATEGORY = "Snow Ice"            [User defined]
>      ORDER = 1                        [This data type is sought first]
>      RUNTIME_PARM_ID = 1509           [Run-time parameter holds LID of alternate]
>      TIMER = "HOURS=6"
>      WAITFOR = "N"                    [Force time-out on wait]
>      TEMPORAL = "N"                   [Use most currently produced]
>   END_OBJECT = ALTERNATE_INPUT
```

Table 11.7.3.2-1. Extract of PGE Metadata ODL File Template Showing Alternate Inputs

```
>END_OBJECT = PCF_ENTRY
>
>OBJECT = PCF_ENTRY
>   CLASS = 17
>   LOGICAL_ID = 1501
>   PCF_FILE_TYPE = 1
>   DATA_TYPE_NAME = "MOD10A1"        [MODIS Level 3 Daily Gridded Snow Cover data set]
>   DATA_TYPE_VERSION = "1"
>   DATA_TYPE_REQUIREMENT = 1
>   SCIENCE_GROUP = ""
>   OBJECT = FILETYPE
>      CLASS = 2
>      FILETYPE_NAME = "Single File Granule"
>   END_OBJECT = FILETYPE
>   INPUT_TYPE = "Alternate"
>   OBJECT = ALTERNATE_INPUT
>      CLASS = 2
>      CATEGORY = "Snow Ice"              [User defined]
>      ORDER = 2                          [This data type is sought last]
>      RUNTIME_PARM_ID = 1509             [Run-time parameter holds LID of alternate]
>      TIMER = "HOURS=6"                  [Wait 6 additional hours]
>      WAITFOR = "N"
>      TEMPORAL = "N"
>   END_OBJECT = ALTERNATE_INPUT
>END_OBJECT = PCF_ENTRY
>
>OBJECT = PCF_ENTRY
>   CLASS = 18
>   LOGICAL_ID = 1502
>   PCF_FILE_TYPE = 1
>   DATA_TYPE_NAME = "MIANTASC"       [MISR Terrestrial Atmosphere and Surface
Climatology]
>   DATA_TYPE_VERSION = "1"
>   DATA_TYPE_REQUIREMENT = 1
>   SCIENCE_GROUP = ""
>   OBJECT = FILETYPE
>      CLASS = 3
>      FILETYPE_NAME = "Single File Granule"
>   END_OBJECT = FILETYPE
```

Table 11.7.3.2-1. Extract of PGE Metadata ODL File Template Showing Alternate Inputs

```
>  INPUT_TYPE = "Alternate"
>  OBJECT = ALTERNATE_INPUT
>     CLASS = 3
>     CATEGORY = "Snow Ice"                [User defined]
>     ORDER = 3                            [This data type is sought last]
>     RUNTIME_PARM_ID = 1509               [Run-time parameter holds LID of alternate]
>     TIMER = ""                           [Don't wait for this one]
>     WAITFOR = "Y"                        [Start anyway]
>     TEMPORAL = "N"
>  END_OBJECT = ALTERNATE_INPUT
>END_OBJECT = PCF_ENTRY
```

11.7.4 Minimum/Maximum Number of Granules Production Rule

The Minimum/Maximum Number of Granules Production Rule makes it possible to specify a range of possible granules for a given input or output for a PGE.

- Inputs.
  - Minimum number of granules the PGE needs for full data coverage.
  - Maximum number of granules for the time period.
- Outputs.
  - Minimum number of outputs that the PGE is expected to produce.
  - Maximum number of outputs that the PGE is expected to produce.

For example, a PGE processes data for every 90-minute interval, has a period of 90 minutes, and takes as input a granule with a period of two hours.

- In many instances one granule of the input will satisfy the PGE.
- In other instances, because of the way the two-hour and 90-minute periods overlap, the PGE needs two input granules to cover the time period.
- Therefore,…
  - Minimum Number of Granules = 1.
  - Maximum Number of Granules = 2.

The Minimum/Maximum Number of Granules Production Rule is different from most production rules because it works for both input and output granules. It allows the PGE to request of a range of inputs (i.e., 1-10 granules), so that it runs with as few as one granule but with as many as ten granules. If a PGE needs at least three granules of a particular input, the minimum number of granules is defined as three and the PGE is not executed until at least three granules are available.

Optional outputs are defined when the Minimum Number of Granules is set to zero. In such cases the PGE can produce none of the particular type of output and still be considered to have executed successfully. If a PGE has a non-zero value for a Minimum Number of Granules associated with an output, and fails to produce any granules of that output type, it is marked as failed.
Figure 11.7.4-1 provides an illustration of the Minimum/Maximum Number of Granules Production Rule. In the example the PGE processes data related to a one-hour period and takes in both Input 1 and Input 2. Since Input 1 has a PERIOD of 1/2 hour, every PGE run requires two Input 1 granules. Input 2 has a PERIOD of 15 minutes, so there are four Input 2 granules for every PGE run.

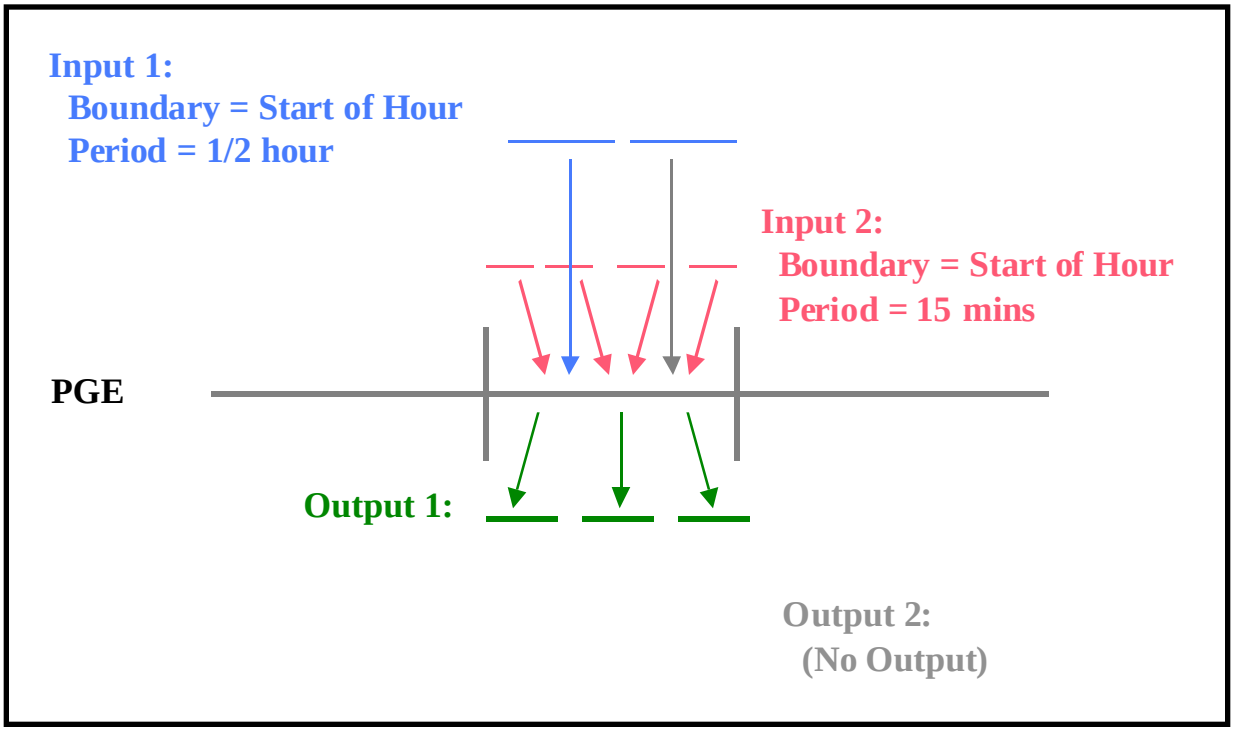

Figure 11.7.4-1. Example of the Minimum/Maximum Number of Granules Production Rule

The PGE produces three Output 1 granules for each run. In this case it does not produce any Output 2 granules.
Minimum and maximum values can affect each input and output as follows:

- Input 1:
  - If Minimum Granules is set to anything equal to or less than two for Input 1, the PGE is scheduled and executed.
  - If Minimum Granules is set to three, the PGE is not scheduled because there are not enough Input 1 granules to make the minimum.
  - If Maximum Granules is set to anything equal to or greater than two for Input 1, the PGE is scheduled and executed.

- If Maximum Granules is set to one, the PGE is not scheduled because there are too many Input 1 granules (the number exceeds the maximum that the PGE can process).

- Input 2:
  - If the Minimum Granules is set to anything equal to or less than four for Input 2, the PGE is scheduled and executed.
  - If Minimum Granules is set to five, the PGE is not scheduled because there are not enough Input 2 granules to make the minimum.
  - If Maximum Granules is set to anything equal to or greater than four for Input 2, the PGE is scheduled and executed.
  - If Maximum Granules is set to three, the PGE is not scheduled because there are too many Input 2 granules (the number exceeds the maximum that the PGE can process).
- Output 1:
  - If Minimum Granules is set to anything equal to or less than three for Output 1, the PGE is scheduled and executes successfully.
  - If Minimum Granules is set to four, the PGE is marked as failed because it did not produce the expected number of output granules.
  - If Maximum Granules is set to anything equal to or greater than three for Output 1, the PGE is scheduled and executes successfully.
  - If Maximum Granules is set to two, the PGE is marked as failed because it produced too many output granules.
- Output 2:
  - If Minimum Granules is set to anything other than zero, the PGE is marked as failed because it did not produce the expected number of output granules.
  - If Maximum Granules is set to anything equal to or greater than zero for Output 2, the PGE is scheduled and executes successfully.

The Minimum/Maximum Granules Production Rules are additions to settings/syntax put into the ODL files for other production rules. All Production Rules have a Minimum and Maximum Granule setting for both inputs and outputs, even though both values may be set to one (1).

11.7.4.1 PGE Science Metadata ODL File Parameters

The PGE science metadata ODL file syntax for implementing the Minimum/Maximum Production Rule for **input** data includes the following types of entries:

```
OBJECT = PCF_ENTRY
   .
   PCF_FILE_TYPE =
   .
   .
   MIN_GRANULES_REQUIRED =
   MAX_GRANULES_REQUIRED =
```

```
                .
                .
                .
        END_OBJECT = PCF_ENTRY
```

Accordingly, the following parameters must be set properly in order to implement the Minimum/Maximum Production Rule:

- PCF_FILE_TYPE.
- MIN_GRANULES_REQUIRED.
- MAX_GRANULES_REQUIRED.

The PCF_FILE_TYPE parameter is defined by integers in the range of 1 to 8 (inclusive). The integers are codes for the following types of files:

- 1 - product input files.
- 2 - product output files.
- 3 - support input files.
- 4 - support output files.
- 5 - user defined runtime parameters.
- 6 - interim/intermediate input files.
- 7 - interim/intermediate output files.
- 8 - temporary input/output.

For inputs (any PCF_ENTRY with a PCF_FILE_TYPE equal to 1, 3 or 6) the following pair of values must be set for each PCF_ENTRY:

- MIN_GRANULES_REQUIRED
  - Minimum number of granules required for the input.
  - A value of zero (MIN_GRANULES_REQUIRED = 0) would mean that the PGE could execute if no granules for that particular input could be found (in effect, the input is an **optional input**).
  - A value of three (for example) would mean that the PGE must have at least three granules of the input before the PGE can be executed.
- MAX_GRANULES_REQUIRED
  - Maximum number of granules for the input that the PGE is able to successfully process.
  - A value of four (for example) would mean that the PGE would process at most four granules for the input.
  - If MAX_GRANULES_REQUIRED = 4 and more than four granules are found for the given input, the PGE is not executed.

The PGE science metadata ODL file syntax for implementing the Minimum/Maximum Production Rule for **output** data includes the following types of entries:

```
        OBJECT = PCF_ENTRY
           .
           PCF_FILE_TYPE =
```

```
.
.
MIN_GRANULE_YIELD =
MAX_GRANULE_YIELD =
.
.
.
END_OBJECT = PCF_ENTRY
```

For outputs (any PCF_ENTRY with a PCF_FILE_TYPE equal to 2, 4 or 7) the following pair of values must be set for each PCF_ENTRY.

- MIN_GRANULE_YIELD
  - Minimum number of granules that the PGE produces for the output.
  - A value of zero (MIN_GRANULE_YIELD = 0) means that the PGE produces no granules for the output (the output is an **optional output**).
  - A value of three (for example) means that the PGE produces at least three granules of the output during a successful execution.
- MAX_GRANULE_YIELD
  - Maximum number of granules that the PGE produces for this output.
  - A value of four (for example) means that at most the PGE produces four granules for the output.
  - Note that sizing of disk space is based on this number, so making it too small could cause problems on the science processor disks.

11.7.5 Optional DPRs Production Rule

The Optional DPRs Production Rule (also called the Data-Scheduled Production Rule) makes the execution of a PGE subject to the availability of a **key input**. The system generates DPRs for every possible instance of the key input data but executes only the DPRs for which data are either produced in data processing or can be acquired from the archive.
The Optional DPRs Production Rule applies to PGEs that process certain kinds of **non-routine data**.

- **Routine Data**
  - Data that can be predicted, that come in at specific intervals and are always of a specified length.
  - Routine data makes it possible for the Basic Temporal Production Rule to schedule PGEs based on their input data.
- **Non-Routine Data**
  - Data that cannot be predicted because they come in at random periods and/or their length is variable.
  - Examples include an "optional" output of an upstream PGE, or data that are archived at random periods (e.g., some forms of ASTER data).

An Optional DPR has as its **key input** a non-routine data type. There are two sets of circumstances that lead to the scheduling of Optional DPRs:

- Every possible time that the input is produced in data processing (i.e., the key input is produced as an "optional" output by an upstream PGE).
- Whenever a new granule (of a particular data type) can be acquired from the archive (e.g., archived data that were inserted at unpredictable times).

An example of the first condition starts with a MODIS PGE that produces a certain product only when the input data were collected during the satellite's "Day" mode. A second MODIS PGE is scheduled to use the optional ("Day"-mode) product from the first MODIS PGE as its key input. The second MODIS PGE is scheduled to run after every instance of the first MODIS PGE; however, only the DPRs that can use the optional products resulting from runs of the first MODIS PGE are executed. The remaining DPRs cannot be executed because there is no input data for them.

The second condition is illustrated by ASTER routine processing, which makes use of the Optional DPRs Production Rule to schedule and execute ASTER PGEs for new data that have been archived. (Note that the DAAC ingests and archives ASTER production data from tapes supplied by the ASTER Ground Data System on a frequent but not entirely predictable basis.) When the Production Planner creates a Production Request for an ASTER PGE, it is necessary to specify the **insertion time** range (i.e., the time period when the desired data were archived) as opposed to the **collection time** (when the satellite instrument gathered the data). DPRs specifying the ASTER PGE are scheduled and executed for the data granules that were actually inserted in the archive during the time period specified in the Production Request.

An illustration of the Optional DPRs production rule is presented in Figure 11.7.5-1. In the figure there are two DPRs (i.e., DPR-1 and DPR-2) for the upstream PGE and two DPRs (i.e., OPT-1 and OPT-2) for the PGE subject to the Optional DPRs Production Rule. The "Optional DPRs" PGE takes as input the optional output of the upstream PGE. When it is executed, DPR-1 produces the optional output, so the dependent DPR (OPT-1) is executed. However, OPT-2 is not executed because DPR-2 (on which OPT-2 depends) does not produce the optional output.

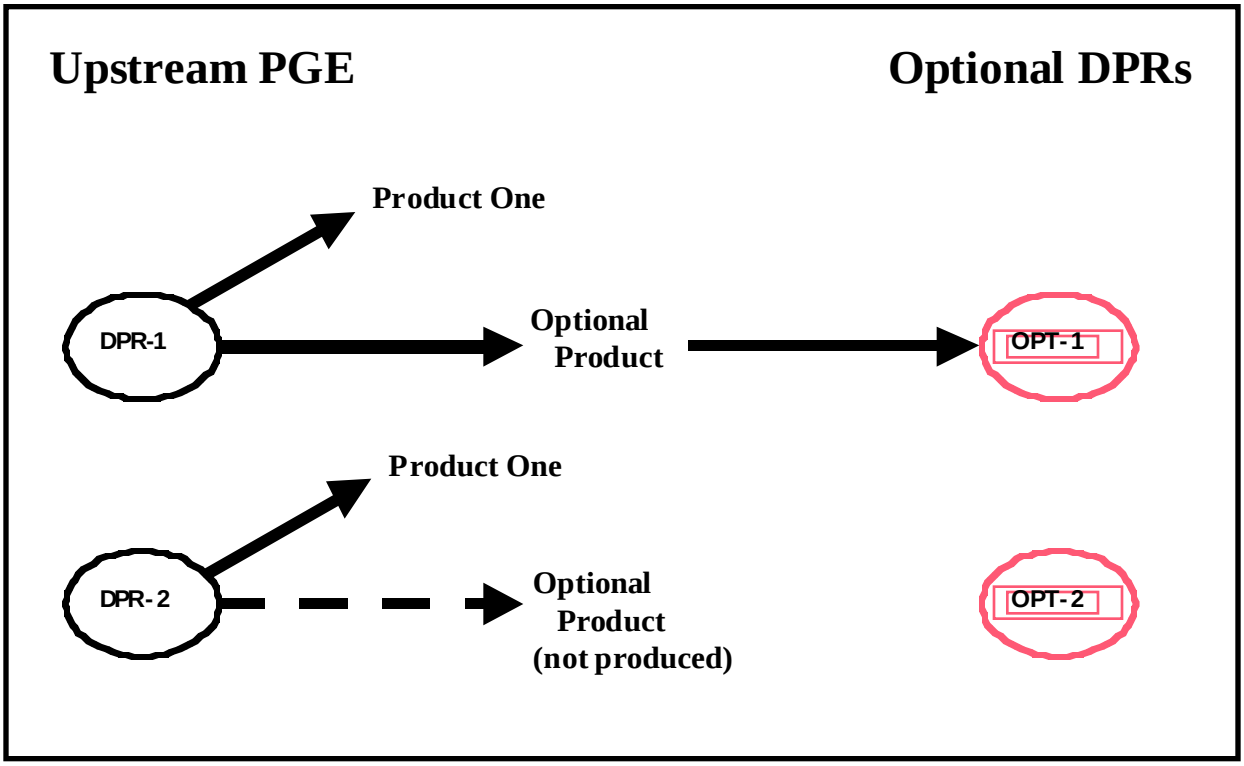

Figure 11.7.5-1 . Example of the Optional DPRs Production Rule

The Optional DPRs Production Rule is set up during the SSI&T process. It uses many of the same parameter settings as the Basic Temporal Production Rule so the values specified in the Basic Temporal Production Rule (or other production rules) are set first, then the Optional DPRs Production Rule syntax is added.

11.7.5.1 PGE Science Metadata ODL File Parameters

The following two types of PGE science metadata ODL file entries must be made in order to set up the Optional DPRs Production Rule:

- SCHEDULE_TYPE.
- KEY_INPUT.

The SCHEDULE_TYPE parameter is set as follows:

- SCHEDULE_TYPE = "Data"
  - This demonstrates the appropriateness of the term "Data-Scheduled Production Rule."
  - Other schedule types include Time, Tile, Orbit, and Snapshot.

The key input is designated by including the following parameter in the PCF_ENTRY for whichever input is to be the key input:

- KEY_INPUT = "Y"
  - Assigning a value of "Y" to the KEY_INPUT parameter identifies the data as a key input and it is subsequently treated as such.
  - Either assigning a value of "N" to the KEY_INPUT parameter or leaving out the parameter entirely identifies the non-key input data.
  - Only one key input is allowed per PGE profile.

The Production Planner's role in the implementation of the Optional DPRs Production Rule was described in the MODIS and ASTER examples previously described and varies with the kind of key input:

- Optional output of an upstream PGE (MODIS example).
  - Production Planner creates Production Requests for the PGE subject to the Optional DPRs Production Rule and specifies the same date/time range as for the upstream PGE.
  - Some of the DPRs generated as a result of the Production Request will never run due to lack of input data.
- Ingested on an irregular time schedule (ASTER example).
  - Production Planner specifies the data **insertion time** range when creating Production Requests.
  - All DPRs generated as a result of the Production Requests should be capable of running.

11.7.6 Intermittent Activation Production Rule

The conditions for executing most PGEs are well defined. The most common activation condition is the availability of all input data sets. Similarly, the frequency of execution is usually well defined (e.g., run once for every granule or run monthly averages once a month). However, some PGEs have additional or different constraints on when they are run.

A PGE can be set up to run on every n^{th} instance of input data. For example, a QA PGE that is run on a daily product may need to be run only every fifth day to provide a spot check. Note that this does **not** refer to the common case of running a weekly averaging PGE only once each week, which would be handled by the Basic Temporal Production Rule and the time ranges specified for the input and output ESDTs. Rather, this is a special case where a PGE **can** be run every day (or hour, week, etc.), but for some reason (such as a QA check) it is desired to run the PGE only every n^{th} day.

To implement the Intermittent Activation Production Rule the Production Planner supplies the following information (via the Production Request Editor) when creating a production request:

- **Number to Skip**
  - Number of DPRs to be skipped (not executed).
  - Entered in the **Skip** field on the Production Request Editor.
- **Number to Keep**
  - After skipping the specified number of DPRs, how many are to be kept?
  - Entered in the **Keep** field on the Production Request Editor.
    - The number to keep is usually one but could be any number.
- **Skip First**
  - Button on the Production Request Editor.
  - Selected to skip the first DPR.
    - Not selected if the first DPR is to be run.

The Planning Subsystem uses the preceding information to establish a pattern of execution. The pattern is effective for the single PR in which the "number to skip" and the "number to keep" are specified; it is not maintained between PRs.

The following example of the Intermittent Activation Production Rule is shown in Figure 11.7.6-1:

- The Production Planner prepares a production request for a 14-day period, generating 14 DPRs.
- The Production Planner made the following selections on the Production Request Editor:
  - Entered "4" in the **Number to Skip** field.
  - Entered "1" in the **Number to Keep** field.
  - Did **not** select the **Skip First** button.
- Consequently, the following results are obtained:
  - First DPR runs. Four DPRs (second through fifth) are skipped.

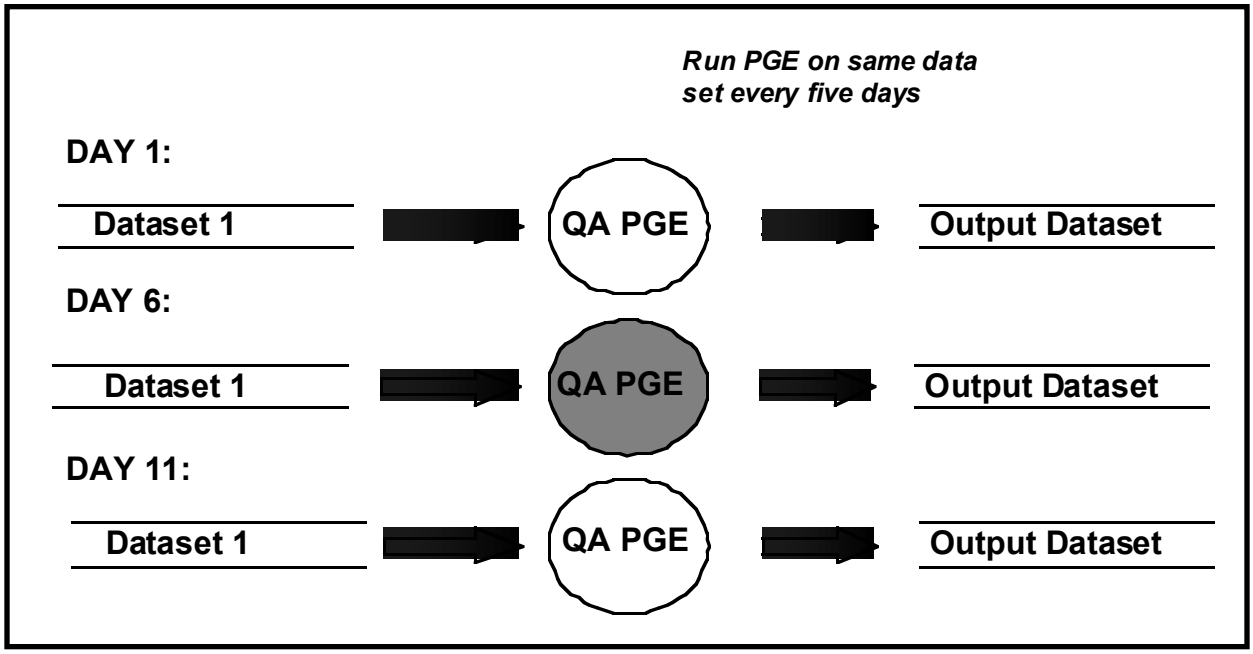

Figure 11.7.6-1. Example of the Intermittent Activation Production Rule

- Sixth DPR runs.
- Four DPRs (seventh through tenth) are skipped.
- Eleventh DPR runs.
- Remaining three DPRs (twelfth through fourteenth) are skipped.

11.7.7 Metadata Checks and Metadata Query Production Rules

The Metadata Checks and Metadata Query Production Rules are similar in definition and use. Both production rules allow the PGE to specify granule-level metadata values that define whether the PGE can accept one (or more) of its inputs. The rules differ only in the results of metadata search performed.

- Metadata Checks Production Rule.
  - When PLS requests the Science Data Server to search for the input(s), the Science Data Server "checks" the metadata of all granules that match the time frame with respect to the value(s) allowed by the PGE.
  - If any granule fails to match the specified value(s), the PGE is not executed.
- Metadata Query Production Rule.
  - When PLS requests the Science Data Server to search for the input(s), the Science Data Server adds to the query the metadata value(s) desired by the PGE.

  - Only the granules that match the time frame of the PGE plus the granule-level metadata value(s) specified by the PGE are staged for the PGE to use as input.
  - If no granules are found matching the conditions and the input is not optional, the PGE is not executed.

- Example of Metadata Checks:
  - A MODIS PGE is run when the Percent Cloud Cover of its inputs is greater than 25 percent.
  - The Metadata Checks Production Rule is used to specify the granule-level metadata value of greater than 25.
  - When the PGE is scheduled and is ready to start, two granules match the timeframe of the Production Request for the input with the Metadata Check.
  - If both granules have a Percent Cloud Cover greater than 25 percent, execution of the PGE starts and both granules are staged.
  - If one of the granules has a Percent Cloud Cover of 15 percent, the PGE is not executed.
- Example of Metadata Query:
  - A MODIS PGE is run when as many granules as possible of one of its inputs have a QA Value = "Good".
  - The Metadata Query Production Rule is used to specify the granule-level metadata value = "Good".
  - When the PGE is scheduled and is ready to start, two granules match the time frame of the production request for the input with the Metadata Query.
  - If both granules have a QA Value = "Good", execution of the PGE starts and both granules are staged.
  - If one of the granules has a QA Value = "Bad", the PGE executes but with only one granule (the one with QA Value = "Good").

The Metadata Checks and Metadata Query Production Rules are used in conjunction with the times specified in the Basic Temporal Production Rule or other production rules. The Metadata Check or Query is added information that further refines what granules are sought by the PGE.
Multi-Granule ESDTs are a special case of the Metadata Query Production Rule. Multi-Granule ESDTs are used for PGE inputs or outputs when more than one granule of the same ESDT exists for the same temporal range (time period). The Multi-Granule ESDT mechanism employs a metadata parameter to differentiate between the "equal in time" granules. A metadata parameter is selected that is unique across granules for the same time period and that is used by PDPS to keep track of which granule is which when the granules are produced. Later, if only one of a pair of granules for a particular time period is needed as input to the PGE, the Metadata Query is used to ensure that PDPS schedules the correct granule as input.
The **Data Day Production Rule** is actually an addition to the Metadata Query Production Rule involving runtime parameter values. There is a pair of settings (Start Data Day and End Data

Day) that allow a PGE to perform a Metadata Query for the start of the Data Day and the end of the Data Day. A separate section of this lesson is devoted to the Data Day Production Rule. Using runtime parameter values is a capability of the Metadata Query and Metadata Checks Production Rules. Rather than use a hard-coded value for the check or query, a value computed from one of the other production rules can be used.
Figure 11.7.7-1 illustrates the Metadata Checks and Metadata Query Production Rules. If no Metadata Check or Query were applicable, the PGE shown in the figure would use three granules of input (i.e., Granules A through C). However, let us assume that the metadata value to be checked/queried is %CloudCover. Each granule has a different value for %CloudCover.

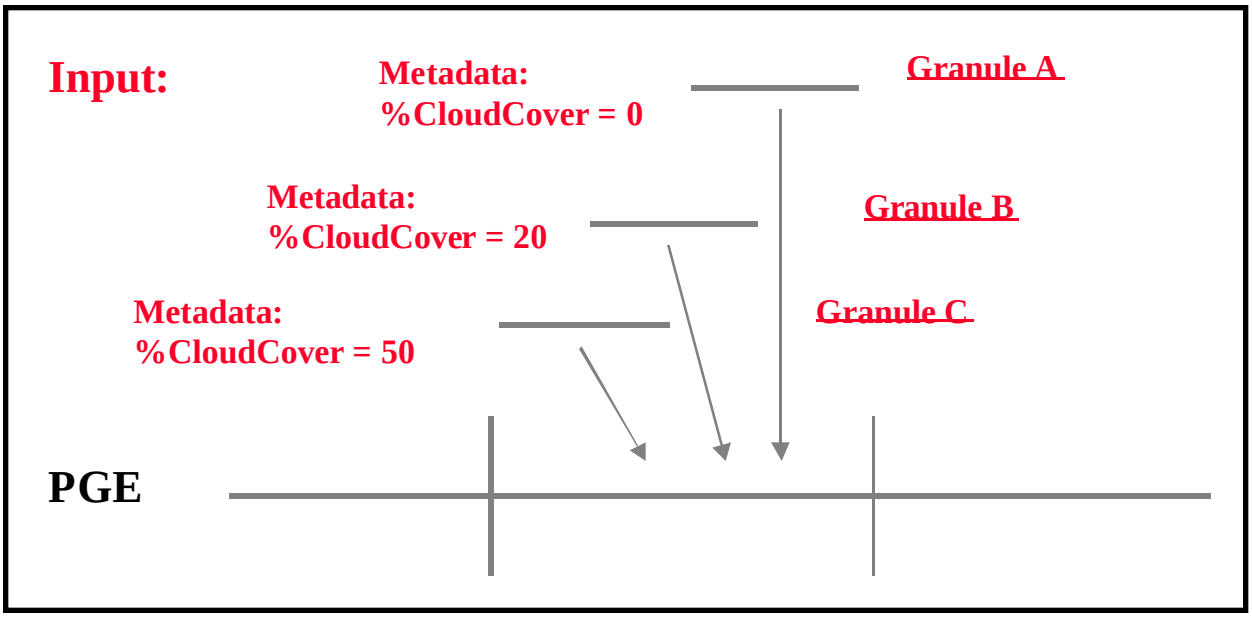

Figure 11.7.7-1. Example of the Metadata Checks and Query Production Rules

The following results demonstrate the differences between the Metadata Checks and Metadata Query Production Rules, especially with respect to the number of inputs that the PGE receives when different values are specified:

- Metadata Check of %CloudCover < 80:
  - In this case all three granules are acquired and the PGE is scheduled and executed.
- Metadata Query of %CloudCover < 80:
  - All three granules are acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover = 50:
  - The PGE is not scheduled because only one of the three granules (Granule C) meets the criterion.
- Metadata Query of %CloudCover = 50:

  - Granule C is found and if the PGE's Min/Max Granules parameters are set to allow one granule, that one granule is acquired and the PGE is scheduled and executed.

- Metadata Check of %CloudCover < 50:
  - The PGE is not scheduled because only two of the three granules (Granule A and B) meet the criterion.
- Metadata Query of %CloudCover < 50:
  - Granules A and B are found and if the PGE's Min/Max Granules parameters are set to allow two granules, the granules are acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover <= 50:
  - The PGE is scheduled and executed because all three granules meet the criterion.
- Metadata Query of %CloudCover <= 50:
  - All three granules are found and acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover = 20:
  - The PGE is not scheduled because only one of the three granules (Granule B) meets the criterion.
- Metadata Query of %CloudCover = 20:
  - Granule B is found and if the PGE's Min/Max Granules parameters are set to allow one granule, the granule is acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover < 20:
  - The PGE is not scheduled because only one of the three granules (Granule A) meets the criterion.
- Metadata Query of %CloudCover < 20:
  - Granule C is found and if the PGE's Min/Max Granules parameters are set to allow one granule, the granule is acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover = 10:
  - The PGE is not scheduled because none of the three granules meets the criterion.
- Metadata Query of %CloudCover = 10:
  - The PGE is not scheduled because no granules are returned from the query (unless Minimum Granules is set to 0).

Note that there can be more than one Metadata Check or Metadata Query on a given input. In the preceding example, a Metadata Check on %CloudCover can be combined with a Metadata Query on another parameter to further limit the input.
The Metadata Checks and Metadata Query Production Rules are additions to settings/syntax put into the ODL files for other production rules. The addition of a Metadata Check or a Metadata Query to an input means that other production rules used to evaluate that input will be applied in combination with the Metadata Check or Metadata Query.

11.7.7.1 PGE Science Metadata ODL File Parameters

Although the Metadata Checks and Metadata Query Production Rules are similar, there are two different ODL objects used to define them within a PCF_ENTRY in the PGE science metadata ODL file; i.e., the METADATA_CHECKS object and the METADATA_QUERY object.
The METADATA_CHECKS object has the following syntax:

```
OBJECT = PCF_ENTRY
   .
   .
   .
   .
   .
   OBJECT = METADATA_CHECKS
      .
      .
      .
   END_OBJECT = METADATA_CHECKS
END_OBJECT = PCF_ENTRY
```

The METADATA_QUERY object has the same syntax except “METADATA_QUERY” replaces “METADATA_CHECKS” in every instance.
Most of the following parameters must be set in the PGE science metadata ODL file within the METADATA_CHECKS or METADATA_QUERY ODL object (as applicable) in order to implement either the Metadata Checks or Metadata Query Production Rule:

- CLASS.
- PARM_NAME.
- OPERATOR.
- VALUE.
- DATABASE_QUERY.
- KEY_PARAMETER_NAME (optional).
- KEY_PARAMETER_VALUE (optional).

CLASS is a simple counter used to differentiate the different Metadata Checks or Metadata Query objects within the file. Since each Metadata Checks or Metadata Query object resides within a different PCF_ENTRY object, the CLASS for an METADATA_CHECKS or METADATA_QUERY object can always be 1 (e.g., CLASS = 1).
PARM_NAME is the name of the metadata parameter on which the check or query is to be performed. The value specified for PARM_NAME (e.g., PARM_NAME = “%CloudCover”) must be part of the granule-level metadata of the ESDT. In addition, it must match the parameter name specified in the ESDT science metadata ODL file.

OPERATOR is the operator (e.g., OPERATOR = “==”) on which the check/query is to be performed. The following values are valid for OPERATOR:

- ">"
  - Value in metadata must be greater than.
- "<"
  - Value in metadata must be less than.
- ">="
  - Value in metadata must be greater than or equal to.
- "<="
  - Value in metadata must be less than or equal to.
- "=="
  - Value in metadata must be equal to.
- "!="
  - Value in metadata must be **not** equal to.

VALUE is the value (e.g., VALUE = 50) against which the metadata parameter (defined by PARM_NAME) is compared (using the operator specified by the OPERATOR parameter). The value for the VALUE parameter should be the type of data (e.g., integer, string) as defined in the ESDT ODL metadata for the parameter.
DATABASE_QUERY indicates whether the value for the Metadata Check or Query should be retrieved from the PDPS database rather than through the use of the VALUE parameter. Specifying DATABASE_QUERY permits **runtime parameter values** to be used for Metadata Query or Metadata Checks. The following values are valid for the DATABASE_QUERY parameter:

- "NONE"
  - Use the value in the VALUE parameter; no value from the PDPS database is used.
- "PATH NUMBER"
  - Use the Path Number (0-233) of the orbit for which the PGE is scheduled.
- "ORBIT NUMBER"
  - Use the Orbit Number of the orbit for which the PGE is scheduled.
- "TILE ID"
  - Use the Tile ID of the current Data Processing Request.
- "START DATA DAY"
  - Use the Start Data Day for the current Data Processing Request.
- "END DATA DAY"
  - Use the End Data Day for the current Data Processing Request.

KEY_PARAMETER_NAME is an optional parameter that is used to specify the container within a multi-container metadata group (i.e., the MeasuredParameters metadata group in most

ESDTs). The KEY_PARAMETER_NAME (e.g., KEY_PARAMETER_NAME = "ParameterName" for metadata checks or qeuries within the MeasuredParameters group) in conjunction with the KEY_PARAMETER_VALUE allows PDPS to determine which container within the multi-container group is to be the object of the check or query. KEY_PARAMETER_NAME is **not** used for product-specific attributes. KEY_PARAMETER_VALUE is an optional parameter that is used to specify the **value** (e.g., KEY_PARAMETER_VALUE = "LandCoverage") for the container within a multi-container metadata group (i.e. the MeasuredParameters metadata group in most ESDTs). The KEY_PARAMETER_VALUE in both the PGE science metadata ODL file and ESDT science metadata ODL file must match.

Multi-Granule ESDTs are created by adding the following parameter to the PCF_ENTRY in the PGE science metadata ODL file:

- DISTINCT_VALUE.

The DISTINCT_VALUE must be set to the value of the metadata parameter that is used to differentiate granules within the Multi-Granule ESDT. In addtion, the input or output defined by the PCF entry must have a corresponding DISTINCT_PARAMETER entry in the ESDT science metadata ODL file.

11.7.7.2 ESDT Science Metadata ODL File Parameters

The METADATA_DEFINITION ODL object surrounds the definition for Metadata Checks or Metadata Query information within the ESDT science metadata ODL file. An OBJECT/END_OBJECT pair is needed to separate the parameters defining the Metadata Definition from the rest of the parameters defining the ESDT with the following syntax:

```
OBJECT = METADATA_DEFINITION
    .
    .
    .
END_OBJECT = METADATA_DEFINITION
```

A METADATA_DEFINITION object can match multiple Metadata Checks or Metadata Query objects in various PGE science metadata ODL files. There is no difference between the two production rules with respect to the parameters that need to be set in the ESDT science metadata ODL file. Most of the following parameters must be set:

- CLASS.
- PARM_NAME.
- CONTAINER_NAME.
- TYPE.
- KEY_PARAMETER_NAME (optional).
- KEY_PARAMETER_VALUE (optional).

CLASS is a simple counter used to differentiate the different Metadata Definition objects within the file. Each Metadata Definition object within the file must have a **different** CLASS value.
PARM_NAME is the name of the Metadata parameter on which the check or query will be performed. The value specified for PARM_NAME must be part of the granule-level metadata of the ESDT. It must also match the parameter name specified in the PGE science metadata ODL file(s).
CONTAINER_NAME is the name of the Metadata Group within which the metadata parameter defined by PARM_NAME is contained. For product-specific attributes CONTAINER_NAME is set to the string "AdditionalAttibutes" (i.e., CONTAINER_NAME = "AdditionalAttibutes").
TYPE indicates the type of data within the metadata parameter. The following values are valid for TYPE:

- "INT"
  - Integer data.
- "FLOAT"
  - Floating point data.
- "STR"

- String or character data.
- Note that dates and times are considered string data.

KEY_PARAMETER_NAME is an optional parameter that is used to specify the container within a multi-container metadata group (i.e., the MeasuredParameters metadata group in most ESDTs). The KEY_PARAMETER_NAME allows PDPS to determine which container within the multi-container group is to be the object of the check or query.
KEY_PARAMETER_VALUE is an optional parameter that is used to specify the value for the container within a multi-container metadata group (i.e., the MeasuredParameters metadata group in most ESDTs). The KEY_PARAMETER_VALUE in both the ESDT science metadata ODL file and PGE science metadata ODL file must match.
The ESDT science metadata ODL file for an input specifying Multi-Granule ESDTs needs to have the following parameter added:

- DISTINCT_PARAMETER.

The DISTINCT_PARAMETER must be set to the name of the metadata parameter that is used to differentiate granules within the Multi-Granule ESDT. A corresponding METADATA_DEFINITION must be created to help PDPS find the specified metadata parameter when querying the Science Data Server.

11.7.8 Data Day Production Rule

The Data Day Production Rule is an addition to the Metadata Query Production Rule involving runtime parameter values. The Data Day Production Rule uses a query to the PDPS database for the time period for the DPR and a Metadata Query for data matching the Data Day. The Data Day is defined as a day within twelve hours of the current day. There is a pair of settings (Start Data Day and End Data Day) that provide parameters for the Metadata Query.

The Start Data Day and End Data Day values are calculated by subtracting twelve hours from the starting day for which the PGE is executing and adding twelve hours onto the ending day for which the PGE is running.

- Data Day for a PGE is running on data from 07/04 00:00:00 to 07/05 00:00:00 is defined as follows:
  - START_DATA_DAY = 07/03 12:00:00
  - END_DATA_DAY = 07/06 12:00:00.

11.7.9 Spatial Query Production Rule

The Spatial Query Production Rule allows a PGE to select input(s) based on the spatial coverage of another input (called the **key input**). The PDPS queries the Science Data Server for the spatial coverage of the key input, then uses it in acquiring any subsequent inputs that the PGE has requested that have the same spatial coverage.

- Example:
  - Level 0 input data for an ASTER DPR covers a small section of the Earth.
  - The PGE requires ancillary data that covers the same area to complete its processing.
  - The PGE uses the Spatial Query Production Rule to mark the geographic input as its key input.
  - The PGE specifies that the ancillary input is to be retrieved for the same spatial coverage as that of the key input.
  - When PDPS finds an input granule for the PGE, it performs a Spatial Query to acquire the ancillary input with the same spatial coverage as that of the key input.

Without specifying coordinates, PDPS can match inputs against the spatial constraint of the key input, and give to a PGE only those granules which overlap in area.
For Release 5B Spatial Pad will be added to the Spatial Query Production Rule. Spatial Pad is a means of padding the spatial constraints of the key input. The specified pad is added to all sides of the key input's spatial shape. All granules that intersect the expanded area are retrieved.
Figure 11.7.9-1 is an illustration of the Spatial Query Production Rule. The figure shows a PGE that has two input types, one of which is the key input. The other type of input has granules labeled with the names of various colors. One granule (i.e., “green”) of the key input is found. The spatial coordinates of the granule are retrieved and all inputs of the second ESDT are checked for overlap with the key input’s coordinates.

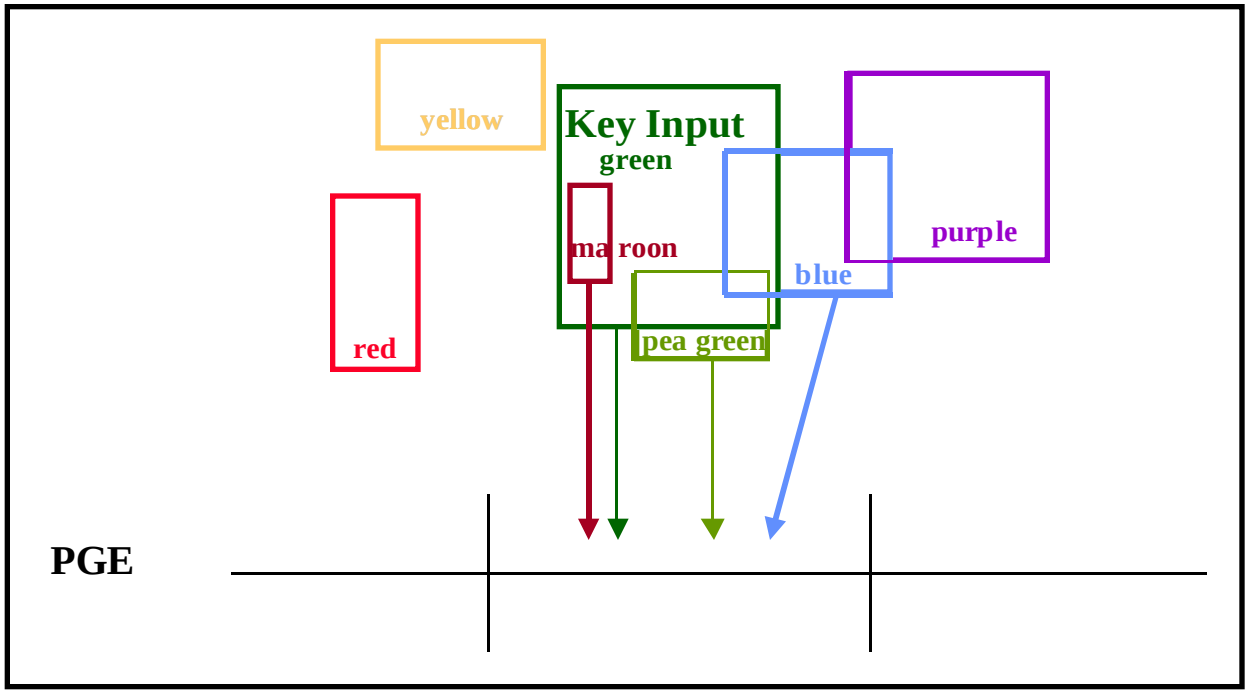

Figure 11.7.9-1. Example of the Spatial Query Production Rule

Assuming that all granules relate to the same time period, the granules are evaluated as follows:

- The “yellow” granule is not retrieved as an input because its spatial coordinates do not overlap with those of the key input.
- The “red” granule is not retrieved as an input because its spatial coordinates do not overlap with those of the key input.
- The “blue” granule is retrieved as an input because its spatial coordinates overlap with those of the key input. Part of its spatial constraint is within the constraint of the key input.
- The “maroon” granule is retrieved as an input because its spatial coordinates overlap with those of the key input. The spatial constraint of this granule is completely within the constraint of the key input.
- The “pea green” granule is retrieved as an input because its spatial coordinates overlap with those of the key input. Part of its spatial constraint overlaps with that of the key input.
- The “purple” granule is not retrieved as an input because its spatial coordinates do not overlap with those of the key input. It does not matter that it overlaps with another input that is accepted (i.e., the “blue” granule).

The Spatial Query Production rule is somewhat of an addition to other production rules. As such, it needs the same parameter settings as the Basic Temporal Production Rule. The values specified in the Basic Temporal Production Rule (or other production rules) are set first, then the Spatial Query Production Rule syntax is added.

11.7.9.1 PGE Science Metadata ODL File Parameters

In order to implement the Spatial Query Production Rule the following two parameters must be defined in the applicable PCF_ENTRY (each input is defined by a separate PCF_ENTRY in the PGE science metadata ODL file):

- KEY _INPUT.
- QUERY_TYPE.

The entries are made in the following format:

```
OBJECT = PCF_ENTRY
  .
  .
  .
  QUERY_TYPE = "Spatial"
  KEY_INPUT = "Y"
  .
  .
  .
END_OBJECT = PCF_ENTRY
```

QUERY_TYPE indicates what type of query is to be done to acquire the input defined by the PCF_ENTRY object. Valid values are as follows:

- "Temporal" - Input is acquired based on time.
  - The Basic Temporal and/or the Advanced Temporal Production Rules is/are used to get the input.
  - "Temporal" is the value that is assumed if the parameter is left out of the PCF_ENTRY object.
- "Spatial" - Input is acquired based on spatial coordinates (as well as time).
  - An input must be designated the key input to be used in determining the spatial constraints of the search.
  - "Spatial" is the value specified for each input that uses the Spatial Query Production Rule.
- "Tile" - Input is acquired by the spatial definition of a tile.
  - Refer to the Tiling Production Rule for additional information.

- "Already Created Tile" - Input is acquired based on the tile ID of an already created tile.
  - Refer to the Tiling Production Rule for additional information.

The KEY_INPUT is the input on which the spatial queries for other inputs will be based. When a KEY_INPUT parameter is assigned a value of "Y" the corresponding input is designated a key input and is treated as such. A value of "N" or leaving out the parameter entirely specifies a non-key input. Only one (1) key input is allowed per PGE Profile.

11.7.10 Tiling Production Rule

The Tiling Production Rule allows a PGE to run over a series of specific geographic locations called "tiles". The tiles are defined before the PGE is scheduled, specifying the longitude and latitude of four points that outline each tile. When the PGE is scheduled, it is scheduled for an entire day, and data is queried based on both a timeframe and the geographic location specified. Each run of the PGE for that day is for a specific tile, and only data that overlap or fit within the geographical coordinates of the tile are staged for the PGE.

- Example:
  - A MODIS PGE is designed to run on data for a specific geographic location every day.
  - The location is expressed as a polygon defined by latitude and longitude coordinates.
  - The MODIS PGE is scheduled every day, and data are retrieved that match the time period (the day for which the PGE is being executed) and some part of it falls within the geographic constraints of the tile.
  - The PGE runs and produces data that define information about the particular tile.

Period and **boundary** are used to specify the timing of input data and provide indications of how often the PGE should be executed. But at least some of the input data are retrieved on the basis of the coordinates defined for the tile on which the PGE is executing. In fact there are really two kinds of tiling:

- The PGE takes in data based on geographic shapes (tiles) and produces an output or outputs for the specified geographical coverage.
- The PGE takes in an already tiled product as input.
  - This form of tiling is more like a Metadata Query using a runtime parameter value to acquire the correct tiled data.

There are some possible future enhancements to the Tiling Production Rule but they have not been scheduled yet:

- **Zonal Tiling** supports tiles that cover a band around the Earth between two given latitudes.
- **Tile Clustering** involves grouping tiles that cover nearby geographic locations together so that data that span the tiles may be staged only once.
  - Intended to improve the performance of Tiling.
  - Also provides for the ability to prioritize one group of tiles over others (so specific geographic outputs are produced before other geographic outputs).

Runtime parameters can be set to the ID of the tile being processed. Since PDPS schedules a Tiling PGE to run once per tile, it can pass the identifier of the tile to the PGE. The identifier can be placed under a specified runtime parameter in the PCF, or it can be used in a Metadata Query for a PGE that would use already tiled data as input.

Figure 11.7.10-1 provides an example of the Tiling Production Rule. The PGE runs once per defined tile. So for every tile in the Tile Scheme a Data Processing Request is created to run using data that match the geographic extent of the tile. The PDPS sends the coordinates of the tiles (e.g., Tiles 1 through 3 in Figure 11.7.10-1) to the Science Data Server when requesting data and acquires only the granules that fall fully or partially within the defined tile.

The PGE itself must be set up to handle the fact that the entire area of the tile may not be covered by available data. In addition, because PDPS does not keep track of tiles once they have been produced, the PGE must set the metadata of the output products so a downstream Tiling PGE can acquire the correct granules for a given tile. The PDPS matches up the granules needed for a downstream PGE via a query to the Data Server Subsystem.

11.7.10.1 Tiling Based on Already Tiled Data

As previously stated, the second form of Tiling concerns PGEs based on tiles that have already been created by other PGEs. Tiling based on already tiled data is really a combination of the Metadata Query Production Rule and the Tiling Production Rule. The latter is used in running the PGE(s) once per tile, just like any other Tiling PGE. The Metadata Query Production Rule is used in acquiring the previously tiled data by querying the Science Data Server for metadata that match the tile ID that is currently being executed. The query depends on the "runtime parameters" function of Tiling to provide the tile ID relevant to the PGE that is currently being executed.

The Tiling Production Rule is based (at least for the PGE science metadata ODL file) on the same fields used for the Basic Temporal Production Rule. A PGE that performs Tiling still needs a **boundary** and **period** and other such parameters. The difference is that values specified for some of the fields provide Tiling information. Furthermore, Tiling requires that a tile scheme be identified in the PGE science metadata ODL file. The tile scheme is defined in a tile science metadata ODL file.

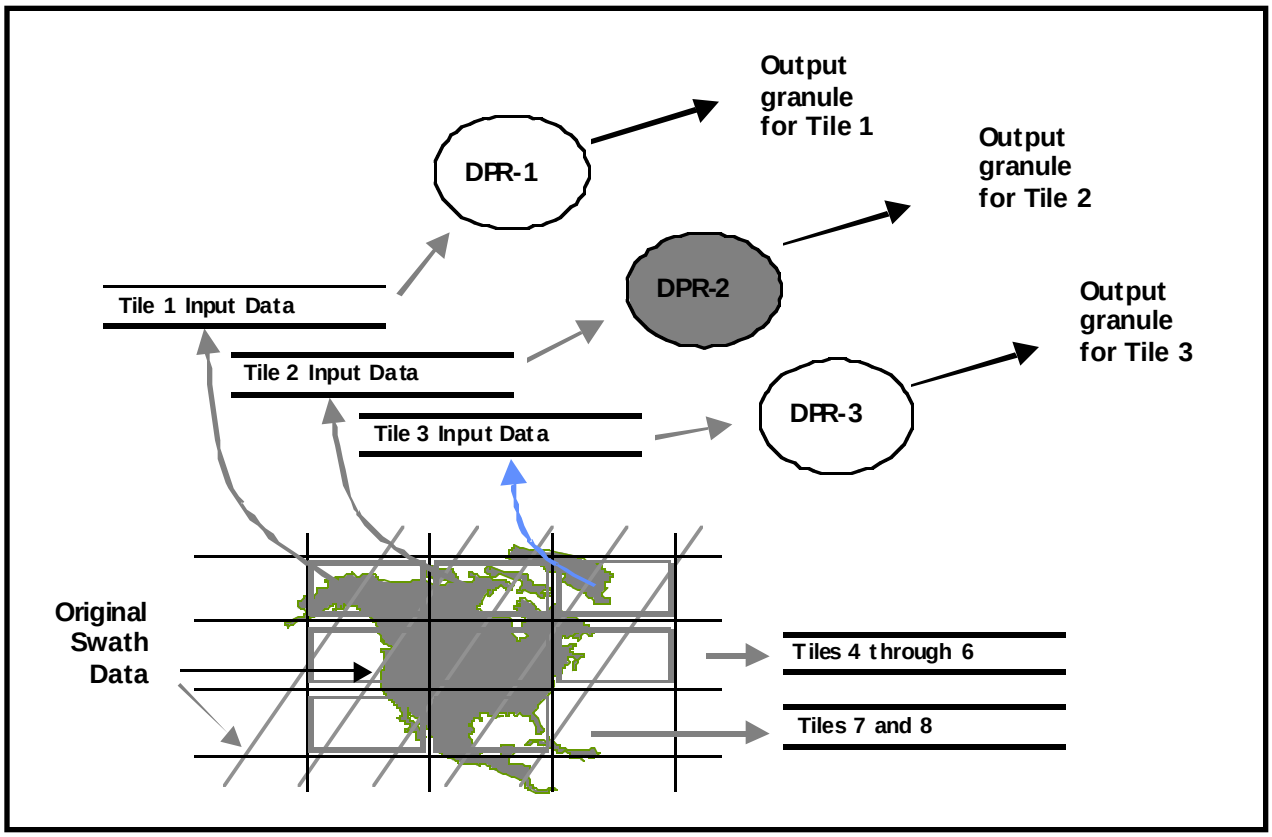

Figure 11.7.10-1. Example of the Tiling Production Rule

11.7.10.2 PGE Science Metadata ODL File Parameters

The following parameters must be set in the PGE science metadata ODL file in order to implement the Tiling Production Rule:

- SCHEDULE_TYPE.
- TILE_SCHEME_NAME.

In addition, the following parameter is used within a PCF_ENTRY when defining the Tiling Production Rule:

- QUERY_TYPE.

The SCHEDULE_TYPE parameter defines the type of scheduling that will be done for the PGE. Values for the Tiling Production Rule are:

- "Tiling"
  - Tile-Scheduled.
  - The PGE is scheduled based on the specified PROCESSING_PERIOD and PROCESSING_BOUNDARY, but a DPR is created for each defined tile.

The TILE_SCHEME_NAME parameter is the name of the Tile Scheme to be used by PDPS when scheduling and executing PGEs for each defined tile. There must be a tile ODL file that matches the specified scheme name.

The QUERY_TYPE parameter specifies the type of query to be performed on the input defined by the PCF_ENTRY Object. It uses the following syntax:

```
OBJECT = PCF_ENTRY
   .
   .
   .
   QUERY_TYPE =
   .
END_OBJECT = PCF_ENTRY
```

For Tiling PGEs there are two possible values for QUERY_TYPE:

- "Tile"
  - The data for the input are acquired on the basis of the spatial constraints of the current tile.
  - Used for a PGE that takes in raw data and produces one or more tiles of data.
- "Already Created Tile"
  - The input is a tiled output of another Tiling PGE.
  - Used for a PGE that takes input from one or more other Tiling PGEs.
  - A Metadata Query must be added to this PCF_ENTRY in order for the correct tiled input to be acquired.

11.7.10.3 Tile Science Metadata ODL File Parameters

The following parameter must be set in the Tile science metadata ODL file in order to implement the Tiling Production Rule:

- TILE_SCHEME_NAME.

In addition, the following ODL objects are used within a PCF_ENTRY to define the Tiling Production Rule:

- TILE object.
- TILE_COORDINATE object.

The TILE_SCHEME_NAME parameter identifies the tile scheme for which the tile information is being specified. Values are limited by the following constraints:

- The string specified can be no more than 20 characters.
- The string specified should match the string specified for TILE_SCHEME in the PGE science metadata ODL file.

The TILE object is an ODL object that surrounds each tile definition. An OBJECT/END_OBJECT pair (as shown in the example that follows) is needed for each tile that is going to be expressly defined:

```
OBJECT = TILE
   .
   .
   .
```

```
END_OBJECT = TILE
```

The following parameters are set in the TILE object in order to implement the Tiling Production Rule:

- CLASS.
- TILE_ID.
- TILE_DESCRIPTION.

CLASS is a simple counter used to differentiate the different TILE objects within the file. Each TILE object needs to have a different CLASS value.

TILE_ID is the tile identifier for the tile being defined. The TILE_ID must be an integer (e.g., TILE_ID = 12) and must be greater than zero but less than the maximum integer. If a Tile ID is defined in other tile schemes, it must have the same coordinates and description.

TILE_DESCRIPTION is a string of characters (255 characters maximum) that describes what the tile is for, such as its geographic location or area that it covers (e.g., TILE_DESCRIPTION = "Upper North America").

The TILE_COORDINATE object is an ODL object that defines a coordinate (latitude and longitude) for a tile. An OBJECT/END_OBJECT pair is needed for each coordinate that is defined. Each tile must have four TILE_COORDINATE objects defined. (Currently only four-sized polygons are allowed; however, a possible future enhancement would provide for polygons with more than four points.) Coordinate objects must follow a clockwise sequence so that if lines were drawn between the points in the order they are given the desired shape would be drawn.

Coordinate objects conform to the following format:

```
OBJECT = TILE
   .
   .        .
   OBJECT = TILE_COORDINATE
      CLASS =
      LATITUDE =
      LONGITUDE =
   END_OBJECT = TILE_COORDINATE
   .
   .        .
END_OBJECT = TILE
```

The following parameters are set in the TILE_COORDINATE object in order to implement the Tiling Production Rule:

- CLASS.
- LATITUDE.
- LONGITUDE.

The CLASS parameter (e.g., CLASS = 1) is an object counter that is used only to distinguish objects. The value assigned to CLASS must be an integer greater than zero and must be unique in the file for the particular type of object.

The LATITUDE parameter (e.g., LATITUDE = 12.15) describes the latitude component of the tile coordinate. There is one LATITUDE entry per TILE_COORDINATE object.

The LONGITUDE parameter (e.g., LONGITUDE = -43.22) describes the longitude component of the tile coordinate. There is one LONGITUDE entry per TILE_COORDINATE object.

11.7.11 Closest Granule Production Rule

The Closest Granule Production Rule allows a PGE to request the nearest input granule from the Data Processing Request time. The PDPS requests a search forward or backward for a specified period of time until it finds a granule that matches the request. However, there is a limit to the number of queries that are performed. The number of queries and the period length of the query are specified during SSI&T.

- Example:
  - A PGE processes data at daily intervals and could use a particular type of calibration granule that would allow it to determine the nearest parameters of the instrument.
  - Although most calibration coefficients are defined as static granules, in this case there is a dynamic granule that is received about once a month.
  - The closest such granule would be optimum, so the PGE uses the Closest Granule Production Rule to search forward or backward from the time of the DPR to find the nearest calibration granule.

The Closest Granule Production Rule supersedes the Most Recent Granule Production Rule. The latter allowed the search for inputs to go backward in time from the start of the DPR. The Closest Granule Production Rule allows the search for input granules to go either backward or forward in time, increasing the flexibility of the rule. The Closest Granule Production Rule has all of the ability of Most Recent Granule, plus the ability to search forward in time for input data. The Closest Granule Production Rule uses two values to determine the period of the query. The two values are concerned with the direction of the query and the number of queries allowed.

- Offset.
  - Tells the PDPS software the query duration.
  - The sign (+ or -) indicates whether the query goes forward (positive) or backward (negative) in time.
- Retries.
  - Tells the PDPS software how many time periods (as defined by the offset) to search (either forward or backward) in time for matching granule.

The PDPS does a Basic Temporal query before using Closest Granule to find the input. If the desired input is not found within the time period of the DPR, PDPS performs a query against the Science Data Server for the period defined by the offset. Again, if no matching granule is found, PDPS repeats the query, going backward or forward in time by the value specified in the offset. If no acceptable granule has been found before the maximum number of queries is reached, PDPS fails to generate the DPR due to insufficient input data.

Figure 11.7.11-1 illustrates the Closest Granule Production Rule. In the example, the PGE has a boundary of “start of day” and a period of one hour, so it is scheduled to run for one hour’s worth of input data. The input has a period of one hour, and can come in at any hour of the day. Consequently, the PGE requests one granule of input.

The PGE has defined the Closest Granule Production rule with a –6-hour period of the query, meaning that it queries back in time in six-hour intervals. The number of retries is two. The PDPS performs a query for the input based on the time period of the DPR. Not finding any matching data, it uses the Closest Granule information to query for a six-hour period beginning six hours before the start time of the DPR. Again nothing is found, so a second Closest Granule

query is performed, this one six hours before the last Closest Granule query. The second query results in the discovery of two matching granules. The PDPS selects the granule that is later in time and schedules the PGE to use it as input.

If the Closest Granule Production Rule were used in conjunction with the Minimum/Maximum Number of Granules Production Rule, it might be possible for both granules to be selected in the previously described Closest Granule query. If the example included setting the Maximum Number of Granules to two, both granules would be selected as input to the PGE.

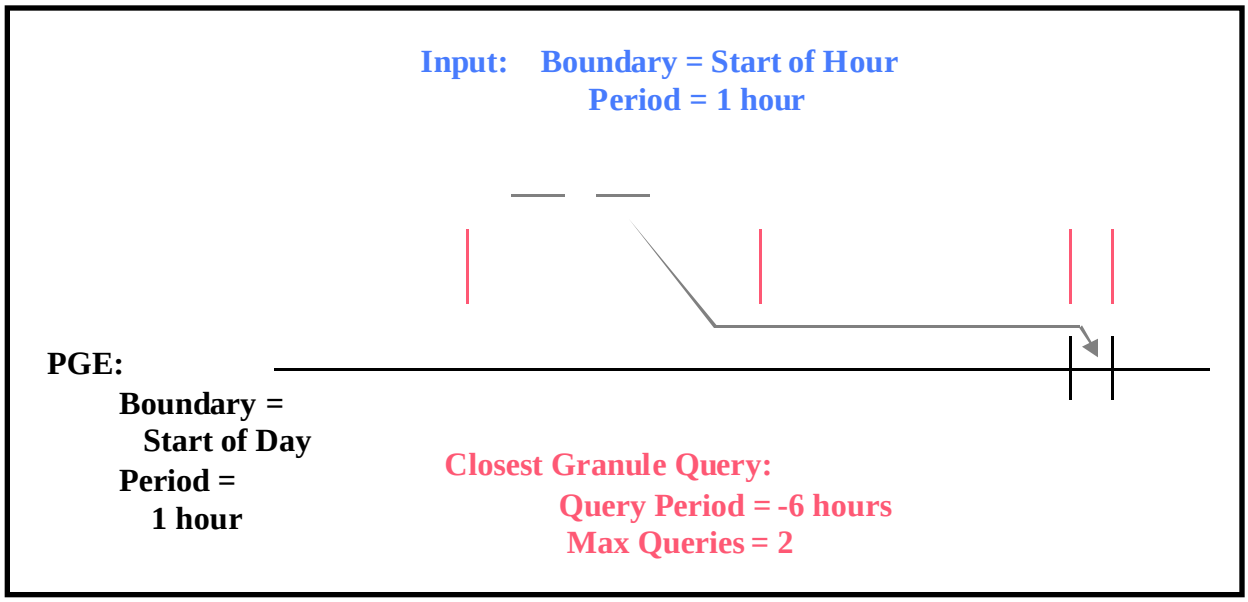

Figure 11.7.11-1. Example of Closest Granule Production Rule

The Closest Granule Production Rule needs the same parameter settings as the Basic Temporal Production Rule. The values needed for the Basic Temporal Production Rule must be set before the Closest Granule Production Rule syntax is added.

11.7.11.1 PGE Science Metadata ODL File Parameters

In addition to the parameter settings for the Basic Temporal Production Rule, the following parameters must be set within the appropriate PCF_ENTRY in the PGE science metadata ODL file in order to implement the Closest Granule Production Rule:

- CLOSEST_QUERY_OFFSET.
- CLOSEST_QUERY_RETRIES.

CLOSEST_QUERY_OFFSET is the offset added to or subtracted from the Data Start Time of the DPR and uses as the query for the requested input data type. The specified value has the format "<Period Type>)=<Length of Period>" (e.g., CLOSEST_QUERY_OFFSET = “HOURS=6”). The following “Period Type” values are used in implementing the Closest Granule rule:

- "WEEKS"
  - Offset is some number of weeks.
  - For example, "WEEKS=2" would be a 14-day offset.
- "DAYS"
  - Offset is some number of days.
  - For example, "DAYS=5" would be a 120-hour offset.
- "HOURS"
  - Offset is some number of hours.

  - For example, "HOURS=4" would be a 240-minute offset.
- "MINS"
  - Offset is some number of minutes.
  - For example, "MINS=5" would be a 300-second offset.
- "SECS"
  - Offset is some number of seconds.

CLOSEST_QUERY_RETRIES is the maximum number of Closest Granule queries before the DPR fails due to insufficient input data. The specified value is an integer value that is limited only by the maximum size of an integer on the executing hardware.
Note that the longer the offset value or the greater the number of retries, the more time that each query requires due to search time at the Science Data Server and processing time of any granules returned. The combination of a large offset with a large number of retries, can (if no data granules are found) consume a lot of time while failing to generate a DPR.

11.7.12 Orbital Processing Production Rule

The Orbital Processing Production Rule is similar to the Basic Temporal Production Rule in that both define the time period for the inputs and outputs of the PGE. The difference is that the Orbital Processing Production Rule uses the orbit of the spacecraft to determine that time period. A PGE that processes data related to every orbit of a satellite uses data related to a time period that is computed from the orbit of that satellite.

- Example:
  - A PGE processes Level 0 data related to each orbit of the AM-1 satellite.
  - The AM-1 satellite has an orbital period of 98 minutes so the PGE is scheduled to process data for each 98-minute interval.
  - Since Level 0 data are received every two hours, the data staged for the PGE include every Level 0 granule that falls within the 98 minute PGE interval.
  - Only one granule of Level 0 data is relevant to some 98-minute orbits.
  - Two granules of Level 0 data are relevant to other 98-minute orbits.

The Orbital Processing Production Rule uses the “period” and “boundary” concept just like the Basic Temporal Production Rule. The difference is that for Orbital Processing, the orbit of the spacecraft is taken into account when a PGE or its data are marked as **orbit scheduled**.
When responding to a Production Request for orbit-scheduled processing, PDPS determines the orbit of the satellite via information provided during the SSI&T process. The information (stored in the PDPS database) gives the start time and length of a particular orbit or set of orbits. PDPS extrapolates (or interpolates in the case of an orbit between two orbital periods stored in the database) the start and end times of the PGE that is specified in the Production Request. Data are sought on the basis of the derived start and stop times and the appropriate data granule(s) is/are staged before the PGE is executed.
Orbital path is the path of the satellite over the Earth. It is a number from 0-233 that indicates the region of the Earth covered by a particular orbit. Note that because of the implementation of Orbital Path, there needs to be a mapping between the orbital path calculated by PDPS and the orbital path number expected by the PGEs.

Runtime parameters can be set to values associated with Orbital Processing. The following list of orbital parameters can be placed under runtime parameters:

- Orbit Number.
  - The number of the orbit (starting from zero) and continually increasing.
- Orbital Path Number.
  - The number of the path that maps to the orbit number.
  - The orbital path number is the 0-233 orbital path traversed by the satellite.
- Orbit Number within the Day.
  - The number of the orbit within the given day.
  - It includes any orbit that starts within the given day.
  - This is a **Release 5B** capability.
- Granule Number within the Orbit.
  - The number of the granule within a given orbit.
  - It includes any granule that starts within the given orbit.
  - This is a **Release 5B** capability.

Figure 11.7.12-1 provides an illustration of the Orbital Processing Production Rule. The PGE in the diagram takes a two-hour input, but is scheduled based on the orbit time and period of the satellite. PDPS uses the data collected at SSI&T to predict the time of the orbit and performs the query to the Science Data Server for the input based on that extrapolated or interpolated orbital time. Granules of input data are allocated to DPRs based on their ability to cover the time period relevant to the DPR.

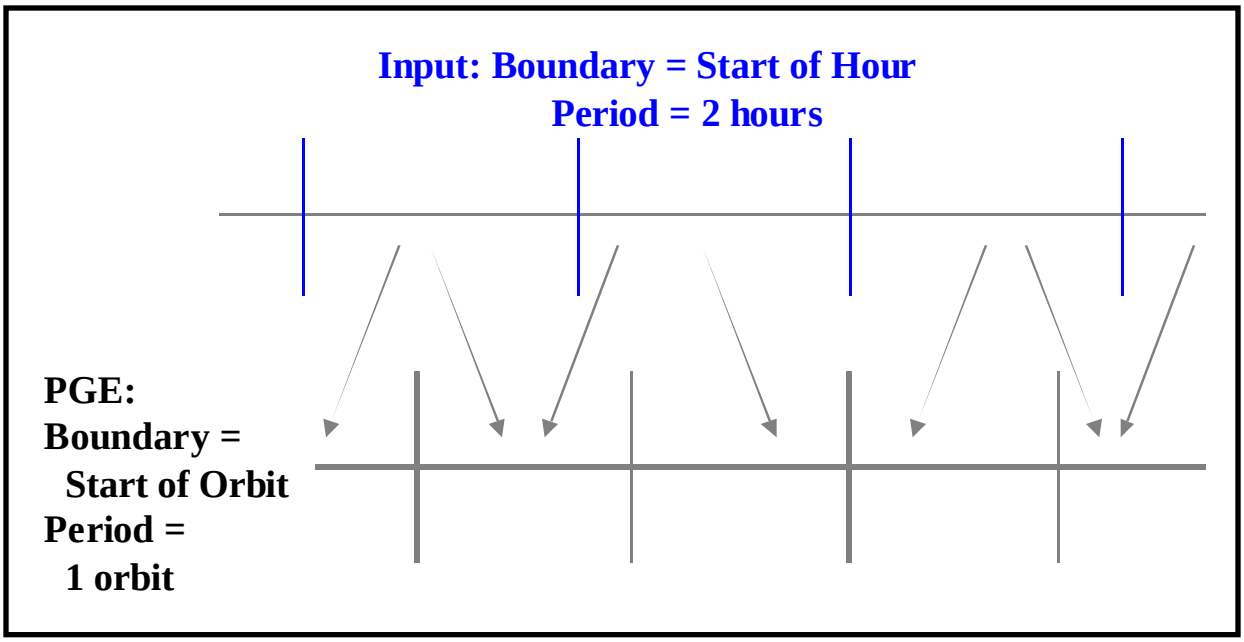

Figure 11.7.12-1. Example of the Orbital Processing Production Rule

In the example shown in Figure 11.7.12-1 the length of an orbit is less than the period of the two-hour input, so sometimes a single granule may cover the input time range of a PGE execution and at other times two granules are required. The production rule would work equally well if the data were of a shorter period (e.g., 1/2 hour) than the orbit of a satellite (e.g., 90 minutes). In such a case three granules would be staged for every execution of the PGE.
The Orbital Processing Production Rule is based (at least for the PGE science metadata ODL file) on the same fields used for the Basic Temporal Production Rule. However, the values specified for the parameters provide orbit information rather than time-period information.

11.7.12.1 PGE Science Metadata ODL File Parameters

The following parameters must be set in the PGE science metadata ODL file in order to implement the Orbital Processing Production Rule:

- PLATFORM.
- SCHEDULE_TYPE.
- PROCESSING_PERIOD.
- PROCESSING_BOUNDARY.

The PLATFORM parameter is the name of the platform (satellite) for which the PGE is processing data. Information concerning the orbits of a satellite are stored in the PDPS database. Values that can be assigned to the parameter are subject to the following constraints:

- The string specified can have no more than 25 characters.
- The string specified should match the string specified in the orbit science metadata ODL file. If no matching file is found, an error is reported during SSI&T.

The SCHEDULE_TYPE parameter describes the type of scheduling that is required for the PGE. "Orbit" is the value used for Orbital Processing. As a result, the PGE is scheduled based on the start time and period of the satellite's orbit. Note that PROCESSING_PERIOD and PROCESSING_BOUNDARY must be set correspondingly.

The PROCESSING_PERIOD is the time interval for the data that the PGE processes. Assuming no combination of production rules that would affect the period, data are acquired for the specified PROCESSING_PERIOD and output data are planned for the given period. The value assigned to PROCESSING_PERIOD is of the format "<Period Type>=<Length of Period>". The "Period Type" applicable to the Orbital Processing Production Rule is "ORBITS". For example, "ORBITS=1" would be applied to a PGE that processes data related to one orbit's worth of data.
The PROCESSING_BOUNDARY is the boundary (starting point in time) of the PGE. It specifies when each instance of the PGE should start. Note that the PROCESSING_BOUNDARY and PROCESSING_PERIOD are used in conjunction when scheduling the PGE. Consequently, "START_OF_ORBIT" is the acceptable PROCESSING_BOUNDARY for the Orbital Processing Production Rule. It indicates that the PGE processes data related to each satellite orbit. It must be used in conjunction with a PROCESSING_PERIOD that specifies a "Period Type" of "ORBITS".

11.7.12.2 Orbit Science Metadata ODL File Parameters

The following parameter must be set in the orbit science metadata ODL file in order to implement the Orbital Processing Production Rule:

- PLATFORM.

In addition, the following ODL object is used in defining orbits for the Orbital Processing Production Rule:

- ORBIT_MODEL object.

The value assigned to the PLATFORM parameter in the orbit science metadata ODL file must be exactly the same as that specified for the same parameter in the PGE science metadata ODL file.
The ORBIT_MODEL object is an ODL object that surrounds each orbit definition. An OBJECT/END_OBJECT pair (as shown in the example that follows) is needed for each orbit that is to be expressly defined: PDPS extrapolates or interpolates orbits that are not specifically defined within the file.

```
OBJECT = ORBIT_MODEL
 CLASS = 1
 ORBIT_NUMBER = 1000
 ORBIT_PATH_NUMBER = 68
 ORBIT_PERIOD = "MINS=98"
 ORBIT_ START = "09/21/1999 14:50:00"
END_OBJECT = ORBIT_MODEL
```

The following parameters are set in the ORBIT_MODEL object in order to implement the Orbital Processing Production Rule:

- CLASS.
- ORBIT_NUMBER.
- ORBIT_PATH_NUMBER.
- ORBIT_PERIOD.
- ORBIT_START.

CLASS is a simple counter used to differentiate the different ORBIT_MODEL objects within the file. Each ORBIT_MODEL object needs to have a different CLASS value.
ORBIT_NUMBER is simply the number of the orbit being specified. Each orbit of the satellite has a sequential number associated with it. This is the integer value of the orbit number for the orbit being defined in the ORBIT_MODEL object.

ORBIT_PATH_NUMBER is value of the path for the specified orbit. The orbital path is a number from 0-233 that repeats every 16 days. This is the integer value of the orbital path number for the orbit being defined in the ORBIT_MODEL object.
ORBIT_PERIOD is the length of time it takes for the satellite to complete one orbit. The value assigned to ORBIT_PERIOD has the format "<Period Type>=<Length of Period>" (e.g., “MINS=98”). Note that the “Length of Period” is specified as a positive integer only.
Period Type values for the orbit model science metadata OFL file are:

- "WEEKS"
  - Orbit spans some number of weeks.
  - For example, "WEEKS=2" would be an orbit that takes two weeks to complete.
- "DAYS"
  - Orbit spans some number of days.
  - For example, "DAYS=5" would be an orbit that takes five days to complete.
- "HOURS"
  - Orbit spans some number of hours.
  - For example, "HOURS=4" would be an orbit that takes four hours to complete.
- "MINS"
  - Orbit spans some number of minutes.
  - For example, "MINS=85" would be an orbit that takes eighty-five minutes to complete.
- "SECS"
  - Orbit spans some number of seconds.
  - For example, "SECS=7200" would be an orbit that takes 7200 seconds (two hours) to complete.

ORBIT_START is the start date and time for the orbit defined by the particular ORBIT_MODEL object. Its format is either "MMM DD YYYY HH:MM:SS" or “MM/DD/YYYY HH:MM:SS”.

11.7.13 Production Planning Considerations

During normal operations it is expected that the Production Planner will not have to add PRs to the PDPS database very frequently. The frequency of this activity is, to some extent, determined by the SCF responsible for the science software.

- The PR is a template request to generate a particular data product and results in a production run of the associated SCF-provided PGE.
  - PR specifies a range (temporal, orbit, or tile) over which the data products are to be produced or the PGEs are to be scheduled.
    - PR might request that the data product be produced for only a single day’s data.

- PR might request that data products be produced for every opportunity of input data for several months, resulting in several hundred jobs being planned and run as the input data become available.

— Early in a mission the SCF may prefer to request processing for a short time period only (e.g., a week or less).

- At that time the SCF is gaining an understanding of the on-orbit behavior of the instrument, the resulting data, and the interaction of the science processing software with real data.

- SCF reviews the quality of the products and notifies the Production Planner of the need for any changes to the PR (e.g., discontinue the PR, change time ranges, or modify input parameters).

— When the SCF has developed a good understanding of the instrument's behavior, the team may be comfortable requesting processing for months at a time.

— DAAC operations may have operational reasons for wanting to issue processing requests for a more limited time period.

- The Production Planner has to balance the various considerations when determining whether or not to create or update a PR.

Planning decisions are made on the basis of locally defined planning strategies for supporting the SCFs' data processing needs. The production planning tools are intended to be flexible enough in their design to support the particular planning and scheduling cycles of the operations organization at each DAAC.

Before planning production the Production Planner must coordinate with the Resource Planner to resolve all resource allocation issues. The Resource Planner notifies the Production Planner of the resources available for use in processing. Furthermore, the Production Planner may well have direct access to the Resource Plan.

The Production Planner prepares monthly and weekly production plans. In addition, the Production Planner develops a daily production schedule from the most current weekly plan. However, the first step in the planning process is creating production requests using the Production Request Editor.